AFOSR-TR-97
0332

PART 53 - FORMS

REPORT DOCUMENTATION PAGE

Form Approveu
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
July, 1996

3. REPORT TYPE AND DATES COVERED
Final Report April 1, 1995 - June 30, 1996

4. TITLE AND SUBTITLE
Multiple Scale Reproducing Kernel Methods for Compressible Flow-Structure Systems

5. FUNDING NUMBERS
AFOSR Grant Number
F49026-95-1-0183

6. AUTHOR(S)
Professor Wing Kam Liu

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Northwestern University Department of Mechanical Engineering
2145 Sheridan Road Evanston, IL 60208

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/ MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research NA Bolling AFB, D.C.
Captain Brian Sanders

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Multiple temporal and spatial scale methods were developed and employed for the study of structural dynamics system in which the response is characterized by multiple-spatial and multiple-time scales. The following fundamental research topics were addressed:
(1) development of multiple scale Reproducing Kernel Particle Methods, which can provide accurate mesh free interpolation functions and possess superior convergence rate for compressible flow-structure interaction problems;
(2) incorporation of multiresolution analysis concepts into RKPM so that the response can be separated into multiple frequency/wave number bands for a better representation of the computed physics;
(3) incorporation of (1) and (2) via a 2D pilot computer code; and
(4) application of the multiple-scale computer code to study nonlinear compressible flow and large deformation structural phenomena.

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

Multiple Scale Reproducing Kernel Methods for Compressible Flow-Structure Systems

April 1, 1995 - June 30, 1996

by

Professor Wing Kam Liu
Principal Investigator

Northwestern University
Department of Mechanical Engineering
2145 Sheridan Road
Evanston, Illinois 60208

AFOSR Grant Number F49620-95-1-0183

Principal Investigator Annual Data Collection (PIADC) Form

PI Name (Last, First, MI) LIU, Wing K.

Institution Northwestern University

Contract/Grant No. AFOSR F49620-95-1-0183

AFOSR USE ONLY
Program Area
Code ______
NX ______
FY 96

Report requested data below-use additional pages if necessary.

A. Publications

Invited W. K. Liu, "Introduction to Wavelet Reproducing Kernel Particle Methods", *USACM Bulletin*, Vol. 8, No. 1, pp. 3-16, 1995.

Invited H. M. Shodja, T. Mura and W. K. Liu, "Multiresolution Analysis of a Micromechanical Model", *Computational Methods in Micromechanics, (Eds. S. Ghosh and M. Ostoja-Starzewski* AMD 212/MD 62 ASME), pp. 33054, 1995.

Invited W. K. Liu, et al. "Overview and Applications of Reproducing Kernel Particle Methods", *Archives of Computational Methods in Engineering, State of the art Review*, Vol. 3, 1, pp. 3-80, 1996.

Invited W. K. Liu, et al. "Multiresolution Reproducing Kernel Particle Method for Computational Fluid Mechanics", *presented at the 3rd US-Japan Symposium on Finite Element Methods in Large-Scale Computational Fluid Mechanics. Accepted for publication in a special issue of International Journal of Numerical Method in Fluids*, 1996.

Invited W. K. Liu, et al. "Advances in Multiple Scale Kernel Particle Methods", *To appear in Computational Mechanics, An International Journal, as a Special Feature Article for the 10th Anniversary Volume of Computational Mechanics*, 1996.

B. Number of researchers

Faculty: 1
Post Docs: 0
Graduate Students: 2
Others: None

C. Honors/Awards (Liu, Wing Kam)

1995 Gustus L. Larson Memorial Award of American Society of Mechanical Engineers, presented at the 1995 International Congress & Exposition, San Francisco, November, 1995.
1995 Fellow of U.S. Association for Computational Mechanics
1995 President Elect of U.S. Association of Computational Mechanics, (Elected) 2000-

1993 Fellow of American Society of Civil Engineers
1990 Fellow of American Society of Mechanical Engineers
1989 Thomas J. Jaeger Prized, International Association for Structural Mechanics in Reactor Technology
1985 Pi Tau Sigma Gold Medal of American Society of Mechanical Engineers
1983 Ralph R. Teetor Educational Award of American Society of Automotive Engineers
1979 Melville Medal of American Society of Mechanical Engineers

D. Interaction/Transitions

a) *Presentations at meetings, conferences, seminars, etc.*

The 3rd U.S. National Congress on Computational Mechanics, Dallas, June, 1995.
ASME PVP Conference, July, 1995.
ASME WAM Meeting, November, 1995.
ARO Workshop, Waterway Experimental Station, April, 1996.

b) *Transitions. Describe cases where knowledge resulting from your effort is used, or will be used, in technology application. DoD etc.*

Argonne, National Laboratory, Dr. Aziz Uras
Dagonet Software, Air Force Phillips Laboratory, 2904 La Veta Dr. NE, Albuquerque, NM 87110-3110, Dr. Lou Baker.
Ontario Hydro Technologies, Ontario, Canada.
The University of Iowa, Dr. J. S. Chen.
The University of Michigan, Dr. G. Hulbert.
Eglin Air Force Base, will visit Dr. Yen Tu, WL/MNMW

ABSTRACT

A novel approach to multiresolution analysis based on reproducing kernel particle methods (RKPM) and wavelets is presented. The concepts of reproducing conditions, discrete convolutions, and multiple scale analysis are described. By means of a newly proposed semi-discrete Fourier analysis, RKPM is further elaborated in the frequency domain, and the interpolation estimate and the convergence of Galerkin solutions are given. The elimination of a mesh, combined with the properties of the dilation and translation of a window function, multiresolution analysis, wavelet-based error estimators, and edge detection brings about a new generation of hp adaptive methods. In addition, this class of multiple scale reproducing kernel particle methods is particularly suitable for problems with large deformations, high gradients, and high modal density. The current application areas of RKPM include structural acoustics, structural dynamics, elastic-plastic deformation, computational fluid dynamics and hyperelasticity.

1. INTRODUCTION

Our emphasis in this paper is on the development of meshless methods for the accurate prediction of the behavior of a complex engineering system that involves a wide spectrum of frequencies and wave numbers. In particular this paper addresses: how does one effectively deal with engineering systems *characterized by multiple temporal and spatial scales*? In addition, can accurate interpolation functions be constructed with the help of *multiresolution analysis concepts* so that the response can be separated into multiple frequency/wave number bands for *a better representation of the computed physics*? Our proposed approach is to employ signal processing theories to attack this difficult problem.

Recently, several different meshless methods have been proposed, including Smooth Particle Hydrodynamics (SPH) method [Monaghan (1988), Gingold and Monaghan (1977), Attaway, Heinstein, Mello and Swegle (1993), Johnson, Peterson and Stryrk (1993), Libersky and Petschek(1990)], Diffuse Elements Method (DEM) by Nayroles, Touzot and Villon (1992), Element Free Galerkin (EFG) [Belytschko, Lu. and Gu (1994a,b, 1995a), Belytschko, Krongauz, Fleming, Organ. and Liu (1995b) and Lu, Belytschko and Gu (1994)], Particle In Cell methods (PIC) [Sulsky, Chen. and Schreyer (1992)], Reproducing Kernel Particle methods (RKPM) [Liu, Adee and Jun (1993b), Liu, Jun, Li, Adee and Belytschko (1995a), Liu, Jun and Zhang (1995b), Liu, Chen, Jun, Chen, Belytschko, Pan, Uras and Chang (1995c), Liu, Jun, Sihling, Chen and Hao (1995d), Liu, Li, and Belytschko (1995e), Liu, Chen and Uras (1995f), Liu (1995) and

Shodja, Mura and Liu (1995) and Liu, Chen and Hao (1996)], wavelet particle methods [Liu and Oberste-Brandenburg (1993a), Liu and Chen (1995)], hp clouds [Duarte and Oden (1995)], partition of unity finite element method [Babuska and Melenk (1995)], finite points method [Oñate, Idelsohn and Zienkiewicz (1995)] and free mesh method [Yagawa, Yamada and Kawai (1995)]. These methods all have unique advantages and disadvantages of their own. On most of these methods, significant advances need to be made before robust treatments of general classes of problems will be possible.

Although some of these methods are currently being used in the design and analysis of large scale engineering systems, it is still in infancy in answering the above raised questions. Even though beautiful computer graphics can be generated to illustrate the numerical results, Fourier type analysis is still required to study the physics of the computed response and to obtain the necessary physical insights. Moreover, since a Fourier spectrum does *not* provide a complete time or space localization information, it is desirable to have a *short time (or space) Fourier spectrum* available at any desirable time and/or spatial location. However, all the computations are performed in time or/and spatial domains. This motivates us to employ a *flexible time-frequency and/or space-wave number window function* to construct global shape functions that can separate the response into different scales (frequencies and wave numbers). With this construction, Fourier type analysis of the computed response is eliminated, since the frequency/wave number bands (or scales) are included in the shape functions.

Our intention here is to introduce multiple scale methods which are based on discrete and continuous reproducing kernels, wavelets, and integral window transforms and to address some of the fundamental issues related to the development of multiple scale meshless methods. It is noted that because of the time/space-frequency/wave number localization via a flexible window function, the interpolation functions are constructed so that the response can be separated into multiple frequency/wave number bands for a better representation of the computed physics.

This class of methods permits the response of a system to be separated into different scales, with wave numbers corresponding to spatial scales and/or frequencies corresponding to temporal scales; so that the response of each scale can be examined separately. Through this multiresolution analysis, the physical interpretation of the computed results can be immediately synthesized, as in classical Fourier analysis; however, without the restrictions of the classical Fourier analysis, which is mostly restricted to simple geometries and discrete frequencies/wave numbers.

This development can be likened to constructing a microscope with a flexible space-wave number and a localized window function which *translates and dilates* in space and time to cover the entire domain of interest. This microscope can magnify, examine, and record the image at various scales and frequencies locally within the support of the window function. The degree of magnification will depend on the power of the microscope, a flexible space-scale and time-frequency window function. This characterization of the response is performed through the integral window transform. Localization can be achieved by contracting the flexible multiple-scale window function. The *zoom in and zoom out* capability of the window function is especially useful in examining complex flow phenomena, such as flow induced vibration, dynamic stability of flow-structure interaction, turbulent structures, structural acoustics, and high frequency structural dynamics response.

Multiple scale methods

Multiple scale analysis [Liu, Zhang and Ramirez (1991), Liu and Haeussermann (1992)] has its origin in signal analysis. Wavelet analysis [Beklkin, Coifman, Daubechies, Mallat, Meyer, Raphael and Ruskai (1992), Chiu (1992), Mallat (1989) and Strang (1989), Williams and Amaratunga (1994) and Daubechies (1992)] is a contemporary science in image processing that has promise in computational mechanics. However, one major drawback in its application to computational mechanics is its inability to handle large deformation and complex boundary conditions. One of the *key* successes of reproducing kernel particle methods (RKPM) [Liu, Jun, Li, Adee and Belytschko (1995a), Liu, Jun and Zhang (1995b), Liu, Chen, Jun, Chen, Belytschko, Pan, Uras and Chang (1995c), Liu, Jun, Sihling, Chen and Hao (1995d), Liu, Li, and Belytschko (1995e), Liu, Chen and Uras (1995f) and Liu, Chen and Hao (1996)] is the formulation of the *boundary correction function* to scaling functions and wavelets. Hence, unlike the usual wavelet and smooth particle hydrodynamics (SPH) analysis, no artificial boundaries are needed in RKPM.

For computational mechanics, a discretization of the system is inevitable. When a system is discretized, aliasing (commonly refers to as high-frequency replicas) is introduced into the response. For a complex system, aliasing may interact with the high frequency part of the response and it becomes impossible to separate them clearly [Liu and Chen (1995)]. In this case, reducing the effect of aliasing is a major step to improving the solution. Moreover, using this local aliasing information, local refinement or hp-like adaptive refinement without a mesh can be carried out without the help of the exact solution.

Liu and Chen (1995) developed an error estimation technique based on multiresolution analysis, which is especially useful for local refinement and convergence studies. The flexible space-scale window function can be constructed to resemble the well-known *unstructured multi-grid and hp-adaptive finite element methods.* However, the multiple scale adaptive refinements are performed simply by inserting nodes into the highest wavelet scale solution region, and at the same time narrowing the window function. Hence, hp-like adaptive refinements can be performed without a mesh. An energy error ratio parameter is introduced as a measure of aliasing error, and critical dilation parameters are determined for a class of spline window functions to obtain optimal accuracy. With this parameter, we are able to separate the numerical noise from the high frequency/wave number component of the physical phenomena. In a traditional numerical analysis, both the numerical noise and the high frequency component of the physics are damped out by the addition of artificial viscosity.

Motivation of multiresolution analysis

A simple illustration of multiresolution analysis is given in Fig. 1.1. The left hand top corner photo presents the highest resolution (256 by 256 pixels). This photo image is digitally separated into a set of consecutive octave (power of 2) scales via a RKPM wavelet transform decomposition (see Sections 7 and 8). The low and high scale images of the original photo are presented below the original photo on the left hand side of Fig. 1.1. As can be seen, the low scale image does not contain any *high frequency components*, whereas, the high scale image captures only the *sharp edges* of the original image. Via this multiresolution analysis, each scale of the image can be studied separately and enhancement or modification can be performed by simply changing the local wavelet coefficients at the desired scale and location.

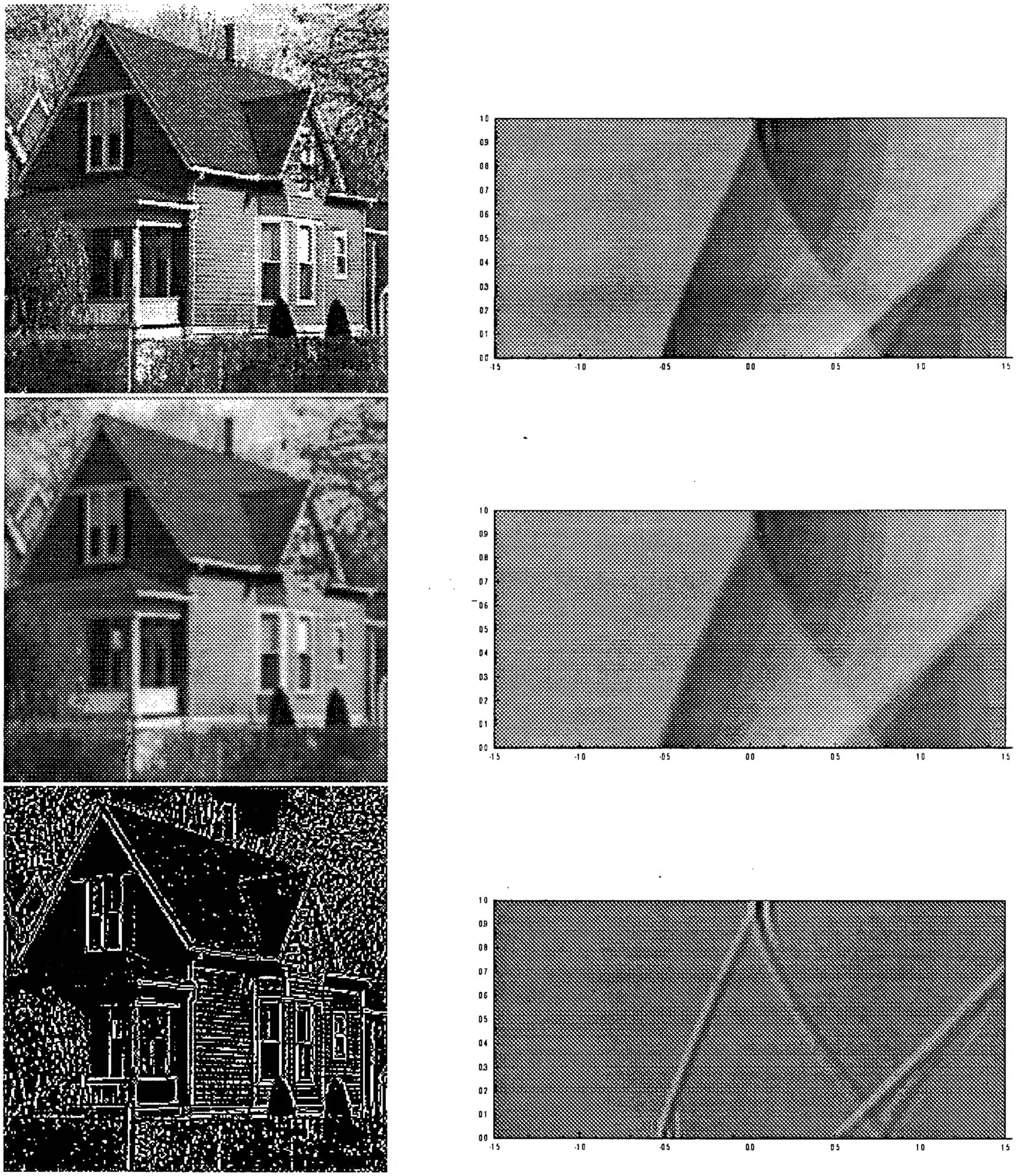


Figure 1.1 Multiresolution RKPM

In a classical Fourier analysis, in which the sine and cosine waves do not have a compact support, such modifications require the computation of a new set of Fourier coefficients which can be very costly. Presently, this is a common practice in the numerical study of the computed physical response. In wavelet analysis, unlike the classical Fourier analysis, only a few wavelet coefficients need to be computed because of the space-scale localization process. In three dimensions, this localization process will involve a space-wave number and time-frequency localized window.

In our research, we borrow this powerful concept of multiresolution analysis to separate the numerical noise from the high scale components (frequencies and wave numbers) of the compressible flow-structure phenomena. Similar to the wavelet decomposition of the photo image presented earlier, the actual computed solution of problem with a discontinuity using the multi-scale RKPM hp-like adaptive refinement [Liu and Chen (1995)], the low scale (scaling function) and the high scale (wavelet) parts of the total solution are presented on the right hand side of Fig. 1.1. As can be seen, the low scale solution consists of the low wave number solution, whereas, the high scale solution consists of a mixture of the high wave number approximation of the physics and the undesirable numerical noise, or the aliasing. Analogous to the low scale photo image, the low scale solution depicts only the smooth part of the solution and the *edges of the discontinuity* (represented by a high wave number wavelet solution) have been removed. However, the *edges of the discontinuity* can be located via this high scale component. This shows that not only the location of the *discontinuity* can be detected, numerical noise can also be minimized by aliasing control.

Section 2 describes the scaling functions and wavelets associated with multiresolution analysis. Two different methods for constructing the Daubechies scaling function are reviewed [Daubechies (1992)]. In Section 3, the orthogonality conditions for the scaling functions and wavelets are studied in the Fourier domain. A systematic procedure of constructing orthogonal scaling functions and orthogonal wavelet functions based on the Fourier transform is also derived. The presentations in Sections 2 and 3 hopefully provide an easier exposition of the construction of the scaling functions and wavelets.

The concepts of scaling functions and wavelets are then used to construct reproducing kernel methods (RKM). The discrete form of a reproducing kernel method is called a reproducing kernel particle method (RKPM) and forms the basis of applying these concepts in computational mechanics. In Section 4, the scaling function of RKM is constructed by relaxing the orthogonality condition. Consequently, a nonuniform discretization of the reproducing kernel can be achieved via a particle method. The concepts involved in the development of RKM from Taylor series, using multi-index notation are introduced in Section 4. An orthogonal window function for RKM is also briefly proposed in the same section. The formulation of RKPM, the discrete form of RKM, is presented in Section 5.

The study of RKM and RKPM formulation in the Fourier space is discussed in Section 6. Interpolant estimates and convergence, and their relationship with the wavelet solution are explored in Section 7. Multiresolution analysis for RKPM is reviewed, and the application of RKPM in edge detection and adaptivity in computational mechanics are outlined in Section 8. The applications of RKPM in large deformation, computational fluid dynamics and structural acoustics are presented in Section 9, followed by conclusions.

2. SCALING FUNCTION AND WAVELETS

2.1 Preliminaries

Multiresolution Analysis and Scaling Function

A multiresolution analysis makes use of a nested sequence of closed subspace $\{V_j\}_{j\in Z}$ and a function ϕ with the following properties

(a). $\cdots \subset V_{-1} \subset V_0 \subset V_1 \subset \cdots \subset L^2(R)$

(b). $\overline{\bigcup_{j\in Z} V_j} = L^2(R)$

(c). $\bigcap_{j\in Z} V_j = \{0\}$

(d). Each subspace is related to its next finer level subspace by

$\phi(x) \in V_j \Leftrightarrow \phi(2x) \in V_{j+1}, \ \ j\in Z$

(e). The translates of $\phi(x)$ span the same subspace so

$\varphi(x) \in V_0 \Leftrightarrow \phi(x - n) \in V_0, \ \ \text{for all } n\in Z$

For an orthogonal wavelet basis, we have the additional condition:

(f). The function $\phi(x - n)$ is a mother function for an orthonormal basis of V_0, where

$$\phi_{j,n} = 2^{j/2}\phi(2^j x - n), \tag{2.1.1}$$

for all $j, n \in Z$, $Z = \{\cdots, -2, -1, 0, 1, 2, \cdots\}$. Conditions (d) and (f) imply $\{\phi_{j,n} \,;\, n \in Z\}$ is an orthonormal basis of V_j for all $j \in Z$.

Since $\phi \in V_0 \subset V_1$, and $\{\phi_{1,n}\}$ forms an orthonormal basis in V_1, any function in V_0 can be expressed in terms of the basis functions of V_1. Thus,

$$\phi(x) = \sum_{n=-\infty}^{\infty} a_n \, \phi(2x - n) \tag{2.1.2}$$

where a_n are given values for each particular function $\phi(x)$, which is often referred to as the "scaling function" of the multiresolution analysis.

Wavelet

A new subspace W_{j-1} associated with the multiresolution analysis is defined as:

$$V_j = V_{j-1} \oplus W_{j-1} \tag{2.1.3}$$

where $\oplus$ is a direct sum, such that W_{j-1} the orthonormal complement of V_{j-1} in V_j, for every $j \in Z$. It follows that for, $j > i$,

$$V_j = V_i \oplus \left(\bigoplus_{k=0}^{j-i-1} W_{i+k} \right), \tag{2.1.4}$$

where all subspaces W_j are orthogonal and that

$$\bigoplus_{j \in Z} W_j = L^2(R) \tag{2.1.5}$$

Basically, for multiresolution analysis, if a sequence of subspaces V_j satisfies the properties (a) - (f), there exists an orthonormal wavelet basis,

$$\psi_{j,n} = 2^{j/2}\psi(2^j x - n),$$

such that , for fixed j, $\{\psi_{j,n}\ ; n \in Z\}$ constitutes an orthonormal basis for W_j. By virtue of (b), (c) and Eq. (2.1.5), this also implies that the whole collection $\{\psi_{j,n}\ ; j,\ n \in Z\}$ is an orthonormal basis for $L^2(R)$. Furthermore, from Eq. (2.1.3), the wavelet function $\psi(x)$ can be expressed in terms of the scaling function $\phi(x)$ at the next finer scale,

$$\psi(x) = \sum_{n=-\infty}^{\infty} b_n\, \phi(2x - n) \tag{2.1.6}$$

Illustration of Multiresolution Analysis by the Haar Function

As an example of a scaling function and associated wavelet, consider the Haar basis. The Haar function, also known as the box function, is given as:

$$\phi(x) = \begin{cases} 1, & 0 \le x < 1 \\ 0, & \text{otherwise} \end{cases} \tag{2.1.7}$$

From Figure 2.1a, it is noted that the box function satisfies Eq. (2.1.2) with coefficients $a_0 = a_1 = 1$ thus

$$\phi(x) = \phi(2x) + \phi(2x - 1) \tag{2.1.8}$$

Furthermore, as shown in Figure 2.1b, the Haar wavelet satisfies Eq. (2.1.6) with coefficients $b_0 = 1$ and $b_1 = -1$, so that:

$$\psi(x) = \phi(2x) - \phi(2x - 1) \tag{2.1.9}$$

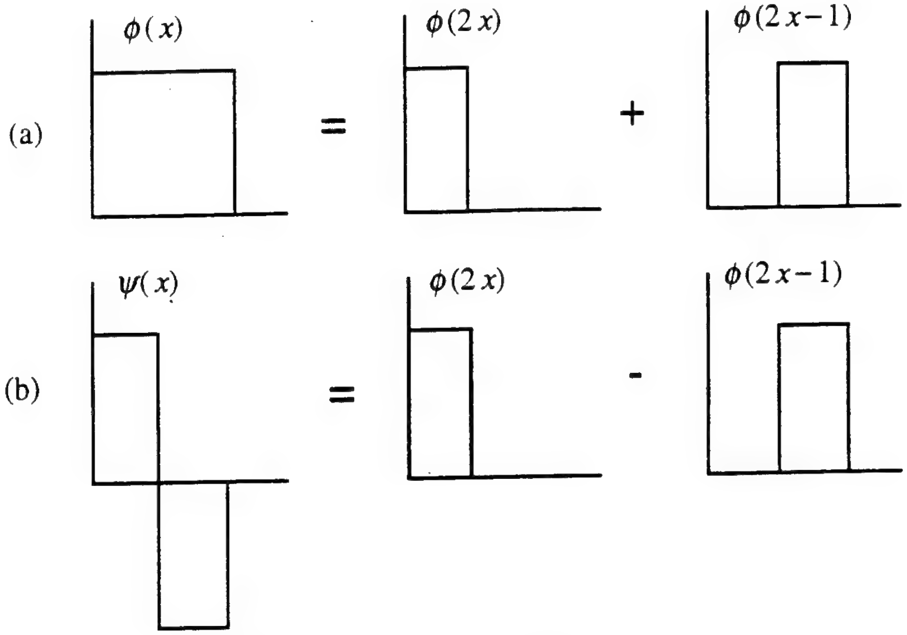


Figure 2.1 Haar Scaling Function and Wavelet

2.2 Daubechies Scaling Function

In order to determine $\phi(x)$ at a set of points, the filter coefficients a_n in Eq. (2.1.2) which satisfy conditions (a) through (e) are required. These conditions are derived from the properties of the scaling function which forms an orthogonal basis.

Conditions for the Determination of Filter Coefficients

(i) The area under the scaling function is normalized to unity to ensure that each scaling function of a given shape is uniquely defined. Thus,

$$\int_{-\infty}^{\infty} \phi(x)\, dx = 1 \qquad (2.2.1)$$

For an N-filter scaling function, Eqs. (2.1.2) and (2.2.1) require that

$$\int_{-\infty}^{\infty} \sum_{n=0}^{N-1} a_n\, \phi(2x - n)dx = \frac{1}{2} \sum_{n=0}^{N-1} a_n \int_{-\infty}^{\infty} \phi(y)\, dy = 1 \qquad (2.2.2)$$

Thus, the first condition on these N-filter coefficients is given as:

$$\boxed{\sum_{n=0}^{N-1} a_n = 2} \tag{2.2.3}$$

(ii) The scaling function has to be orthogonal to its integer translates, that is:

$$\int_{-\infty}^{\infty} \phi(x)\ \phi(x+k)\ dx = \delta_{0k}, \quad k = 0, 1, \ldots, N/2-1 \tag{2.2.4}$$

where δ_{0k} is the Kronecker Delta; in the orthogonality condition, N is restricted to be an even integer. Substituting Eq. (2.1.2) into Eq. (2.2.4):

$$\int_{-\infty}^{\infty} \sum_{n=0}^{N-1} a_n \phi(2x-n) \sum_{\bar{n}=0}^{N-1} a_{\bar{n}} \phi(2x+2k-\bar{n})\ dx = \delta_{0k}, \quad k = 0, 1, \ldots, N/2-1 \tag{2.2.5}$$

and rearranging the indices for $\bar{n} = m + 2k$ yield:

$$\int_{-\infty}^{\infty} \sum_{n=0}^{N-1} a_n \phi(2x-n) \sum_{m=-2k}^{N-1-2k} a_{m+2k} \phi(2x-m)\ dx = \delta_{0k}, \quad k = 0, 1, \ldots, N/2-1 \tag{2.2.6}$$

From Eq. (2.2.4), Eq. (2.2.6) results in:

$$\sum_{n=0}^{N-1} a_n \sum_{m=-2k}^{N-1-2k} a_{m+2k}\ \delta_{nm} = 2\delta_{0k}, \quad k = 0, 1, \ldots, N/2-1 \tag{2.2.7}$$

Therefore, the orthogonality condition in Eq. (2.2.7) provides $N/2$ equations:

$$\boxed{\sum_{n=0}^{N-1} a_n\, a_{n+2k} = 2\delta_{0k}, \quad k = 0, 1, \ldots, N/2-1} \tag{2.2.8}$$

Up to this point, we have $N/2 + 1$ equations to define N coefficients, so another $N/2 - 1$ equations are necessary to determine a unique set of filter coefficients. The approach used is to require the scaling function to be able to represent polynomials of order up to $N/2 - 1$ exactly. This constraint leads to the compactly supported wavelets [Daubechies (1992)].

(iii) Consider a given polynomial function with order $N/2$:

$$f(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \ldots + \alpha_{N/2-1} x^{N/2-1} \tag{2.2.9}$$

This function can also be represented by the following expansion:

$$f(x) = \sum_{n=0}^{N-1} c_n\, \phi(x - n) \tag{2.2.10}$$

The wavelet can be used to obtain conditions on the filter coefficients by taking the inner product of Eq. (2.2.10) with $\psi(x)$, thus:

$$\langle f(x),\, \psi(x)\rangle = \sum_{n=0}^{N-1} c_n \langle \phi(x - n),\, \psi(x)\rangle = 0 \tag{2.2.11}$$

This is because $\phi(x - n)$ is required to be orthogonal to the wavelet function, $\psi(x)$. Hence, using Eq. (2.2.9) in the left hand side of Eq. (2.2.11) gives:

$$\alpha_0 \int_{-\infty}^{\infty} \psi(x)dx + \alpha_1 \int_{-\infty}^{\infty} x\psi(x)dx + \ldots + \alpha_{N/2-1} \int_{-\infty}^{\infty} x^{N/2-1}\psi(x)dx = 0 \tag{2.2.12}$$

Eq.(2.2.12) holds for arbitrary α_a so that the additional $N/2$ conditions are:

$$\int_{-\infty}^{\infty} x^k \psi(x)dx = 0, \quad k = 0, 1, 2, \ldots, N/2 - 1 \tag{2.2.13}$$

To relate Eq. (2.2.13) with the filter coefficients a_n, the Fourier domain is used. The Fourier transform of $\phi(x)$, gives:

$$\mathcal{F}[\phi(x)] = \hat{\phi}(\omega) = \frac{1}{2}\sum a_n e^{in\omega/2}\hat{\phi}(\frac{\omega}{2}) = P(\frac{\omega}{2})\hat{\phi}(\frac{\omega}{2}) \tag{2.2.14}$$

Eq. (2.2.14) can be expressed as:

$$= \left[\prod_{j=1}^{N} P(\frac{\omega}{2^j})\right] \hat{\phi}(\frac{\omega}{2^N}) \cong \prod_{j=1}^{\infty} P(\frac{\omega}{2^j}) \tag{2.2.15}$$

where $P(\omega) = \frac{1}{2}\sum a_n e^{in\omega}$. For an N-filter, a wavelet function can be expressed as [Strang (1989)]:

$$\psi(x) = \sum_{n=2-N}^{1} (-1)^n a_{1-n}\phi(2x-n) \tag{2.2.16}$$

Then the Fourier transform of Eq. (2.2.16) gives:

$$\hat{\psi}(\omega) = \frac{1}{2}\sum_{n=2-N}^{1} (-1)^n a_{1-n} e^{in\omega/2}\hat{\phi}(\frac{\omega}{2}) \tag{2.2.17}$$

Recall the zero moment in Eq. (2.2.13),

$$\int_{-\infty}^{\infty} \psi(x)dx = \hat{\psi}(0) = 0 \tag{2.2.18}$$

When $\omega = 0$, by using Eq. (2.2.18), Eq. (2.2.17) yields:

$$\hat{\psi}(0) = \frac{1}{2}\sum_{n=2-N}^{1} (-1)^n a_{1-n}\hat{\phi}(0) = \frac{1}{2}\sum_{n=2-N}^{1} (-1)^n a_{1-n} = 0 \tag{2.2.19}$$

Note that $\hat{\phi}(0) = 1$, which can be easily verified from the Fourier transform of Eq. (2.2.1). From the definition of $P(\omega)$, we observe that $P(\omega = \pi) = \frac{1}{2}\sum(-1)^n a_n$. Substituting this relation into Eq. (2.2.19) leads to $P(\pi) = 0$ since $\hat{\psi}(0) = 0$. For the *k*th moment in Eq. (2.2.13), it yields:

$$\int_{-\infty}^{\infty} x^k \psi(x)dx = (i)^k \hat{\psi}^{(k)}(0) = 0 \qquad (2.2.20)$$

The *k*th derivative of Eq. (2.2.17) at $\omega = 0$ gives:

$$\hat{\psi}(0)^{(k)} = \frac{1}{2}\left(\frac{i}{2}\right)^k \sum_{n=2-N}^{1} (-1)^n a_{1-n} n^k = 0 \qquad (2.2.21)$$

From Eq. (2.2.21) and the *k*th derivative of $P(\omega)$, $P(\omega)^{(k)} = \frac{i^k}{2}\sum n^k a_n e^{in\omega}$, it shows that:

$$P^{(k)}(\pi) = 0 \qquad (2.2.22)$$

Thus, the third condition in terms of the a_n can be obtained from Eq. (2.2.22):

$$\boxed{\sum_{n=0}^{N\text{-}1} (\text{-}1)^n a_n n^k = 0, \quad k = 0, 1, 2, ..., N/2 - 1} \qquad (2.2.23)$$

It is noted that the equation with $k = 0$ in Eq. (2.2.23) is redundant because it can also be found from the combination of Eqs. (2.2.3) and (2.2.8). Thus, by excluding this redundant equation, Eq. (2.2.23) provides only *N*/2 - 1 new equations. The scaling function and wavelet are uniquely defined by the coefficients a_n which are obtained by solving the system of *N* equations.

An example of filter coefficient for N = 4, the Daubechies D4 scaling function

$$a_0 + a_1 + a_2 + a_3 = 2 \qquad \text{(from Eq. (2.2.3))} \qquad (2.2.24a)$$

$$a_0^2 + a_1^2 + a_2^2 + a_3^2 = 2 \qquad \text{(from Eq. (2.2.8) } k = 0\text{)} \qquad (2.2.24b)$$

$$a_0a_2 + a_1a_3 = 0 \qquad \text{(from Eq. (2.2.8) } k = 1) \qquad (2.2.24c)$$

$$a_0 - a_1 + a_2 - a_3 = 0 \qquad \text{(from Eq. (2.2.23) } k = 0) \qquad (2.2.24d)$$

$$-a_1 + 2a_2 - 3a_3 = 0 \qquad \text{(from Eq. (2.2.23) } k = 1) \qquad (2.2.24e)$$

In this example, the redundant equation Eq. (2.2.24d) can be obtained by:

$$\sqrt{2 * Eq.(2.2.24b) + 4 * Eq.(2.2.24c) - Eq.(2.2.24a)^2} \qquad (2.2.25)$$

The filter coefficients, can easily be obtained by replacing the nonlinear equation $a_0a_2 + a_1a_3 = 0$ with the linear equation $a_0 - a_1 + a_2 - a_3 = 0$, which is the redundant equation. The filter coefficients obtained from Eq. (2.2.24a-e) are also known as the coefficients for the Daubechies D4 scaling function. Table 2.1 lists the filter coefficients for the Daubechies D4

n	a_n
0	0.68301270189222
1	1.18301270189222
2	0.31698729810778
3	-0.188301270189222

Table 2.1 Filter coefficients for the Daubechies scaling function D4

2.3 Construction of Scaling Function

Given a set of filter coefficients as defined in Section 2.2, the corresponding scaling function can be constructed by the iteration method or by the recursion method.

2.3.1 The Iteration method

The iteration method starts from Eq. (2.1.2):

$$\phi_i(x) = \sum_{n=-\infty}^{\infty} a_n \, \phi_{i-1}(2x - n) \qquad (2.3.1)$$

with the box function as $\phi_0(x)$, where $\phi_0(x) = \begin{cases} 1, & 0 \le x \le 1 \\ 0, & otherwise \end{cases}$

Example 1 : $a_0 = 2$

For $a_0 = 2$, the iteration equation obtained from Eq. (2.1.2) yields:

$$\phi_i(x) = 2\ \phi_{i-1}(2x) \qquad \text{for} \quad i \ge 1$$

The Delta function constructed by iteration method with $a_0 = 2$ is depicted in Fig2.2. It is obtained when $i \to \infty$.

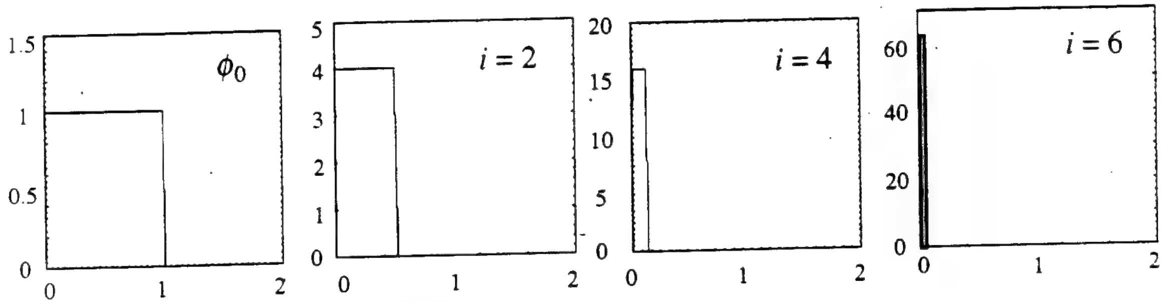


Figure 2.2 The Delta function constructed by iteration with $a_0 = 2$

Example 2 : $a_0 = \frac{1}{2}$, $a_1 = 1$, $a_2 = \frac{1}{2}$

For $a_0 = \frac{1}{2}$, $a_1 = 1$, $a_2 = \frac{1}{2}$, the iteration equation is given as:

$$\phi_i(x) = \frac{1}{2}\ \phi_{i-1}(2x) + \ \phi_{i-1}(2x\text{-}1) + \frac{1}{2}\ \phi_{i-1}(2x\text{-}2) \qquad \text{for} \quad i \ge 1$$

The hat function constructed by iteration method with $a_0 = \frac{1}{2}$, $a_1 = 1$, $a_2 = \frac{1}{2}$ is shown in Fig. 2.3. It is obtained when $i \to \infty$.

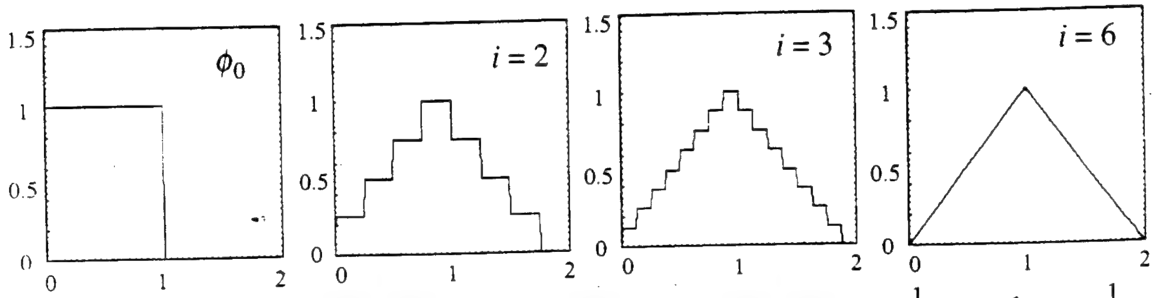


Figure 2.3 The Hat function constructed by iteration with $a_0 = \frac{1}{2}$, $a_1 = 1$, $a_2 = \frac{1}{2}$

Example 3: D4 filter coefficients

The coefficients in Eq. (2.2.21), lead to the D4 scaling function. The plot of the D4 scaling function as constructed by iteration is shown in Fig. 2.4b.

Remark: The iteration method is simple and straight forward, although, it is not computationally efficient.

2.3.2 The Recursion Method

The crux of the recursion method hinges on the fact that the scaling function $\phi(x)$ is known at integer points, $x = j$. Thus, recursion gives the values of $\phi(x)$ at the half integer points and at the quarter integer points for the next recursion. Repeating this process will yield all dyadic points $(2^j, j \geq 0)$. This leads to a fast algorithm for wavelet calculations. To compute $\phi(x)$ at the integer points, consider the expansion of Eq. (2.1.2):

$$\phi(x) = a_0\phi(2x) + a_1\phi(2x - 1) + \ldots + a_{N-1}\phi(2x - N + 1) \tag{2.3.2}$$

For N point values of $\phi(x)$, Eq. (2.3.2) can be written in matrix form:

$$\begin{bmatrix} a_0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ a_2 & a_1 & a_0 & 0 & \cdots & 0 & 0 \\ a_4 & a_3 & a_2 & a_1 & a_0 & \cdots & 0 \\ a_6 & a_5 & a_4 & \cdots & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & a_{N-1} & a_{N-2} & a_{N-3} \\ 0 & 0 & 0 & \cdots & 0 & 0 & a_{N-1} \end{bmatrix} \begin{bmatrix} \phi(0) \\ \phi(1) \\ \phi(2) \\ \phi(3) \\ \vdots \\ \phi(N-2) \\ \phi(N-1) \end{bmatrix} = \begin{bmatrix} \phi(0) \\ \phi(1) \\ \phi(2) \\ \phi(3) \\ \vdots \\ \phi(N-2) \\ \phi(N-1) \end{bmatrix} \tag{2.3.3}$$

or

$$(\mathbf{M} - \mathbf{I})\,\mathbf{\Phi} = \mathbf{0} \tag{2.3.4}$$

The vector values of $\mathbf{\Phi}$ at the integer points is the eigenvector of $\mathbf{M}$ corresponding to the eigenvalue 1. As in all eigenvalue problems, this vector is not uniquely defined. An additional normalizing condition obtained from Eq. (2.2.1) is used to determine a unique eigenvector. Thus,

$$\sum_{n=-\infty}^{\infty} \phi(n) = 1, \qquad n \in Z \tag{2.3.5}$$

Once we know the values of $\phi(x)$ at the integer points, then the values of $\phi(x)$ at the half integer points can be computed by Eq. (2.1.2):

$$\phi(\tfrac{x}{2}) = \sum_{n=-\infty}^{\infty} a_n \, \phi(x - n) \qquad (2.3.6)$$

Example 4: D4 scaling function

As an example the recursion method, the values of D4 at four integer points are calculated by Eq (2.3.3):

$$\begin{bmatrix} a_0 & 0 & 0 & 0 \\ a_2 & a_1 & a_0 & 0 \\ 0 & a_3 & a_2 & a_1 \\ 0 & 0 & 0 & a_3 \end{bmatrix} \begin{bmatrix} \phi(0) \\ \phi(1) \\ \phi(2) \\ \phi(3) \end{bmatrix} = \begin{bmatrix} \phi(0) \\ \phi(1) \\ \phi(2) \\ \phi(3) \end{bmatrix}$$

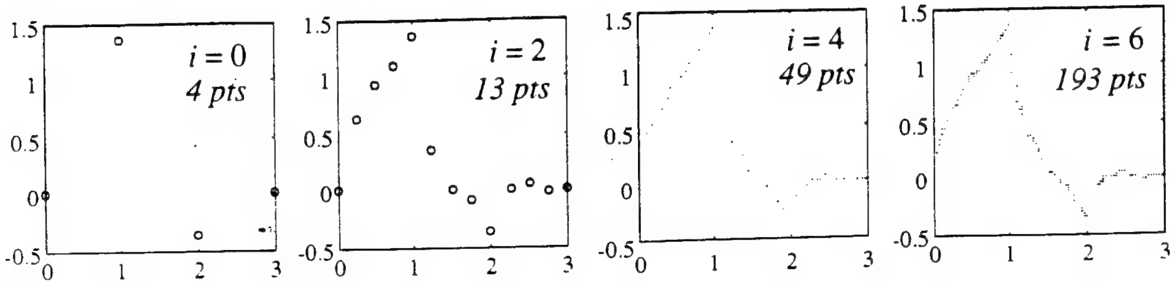
The results are:

$$\phi(0) = 0, \;\; \phi(1) = \frac{1 + \sqrt{3}}{2}, \;\; \phi(2) = \frac{1 - \sqrt{3}}{2}, \;\; \phi(3) = 0$$

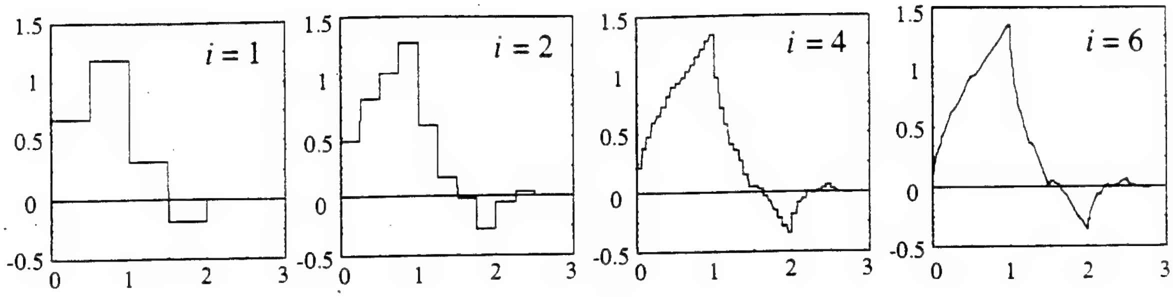
Then, from Eq. (2.3.6), the values of $\phi(x)$ at the half integer points are given:

$$\phi(\tfrac{1}{2}) = \frac{2 + \sqrt{3}}{4}, \;\; \phi(\tfrac{3}{2}) = 0, \;\; \phi(\tfrac{5}{2}) = \frac{2 - \sqrt{3}}{4}$$

A comparison of the D4 scaling functions obtained by iteration and recursion are shown in Figure 2.4. Apparently, the construction from recursion is more efficient and accurate than from iteration.



(a)



(b)

Figure 2.4 The D4 scaling function constructed by recursion (a) and by iteration (b)

Derivative of Scaling Functions

The derivative of the scaling function can be computed by differentiating Eq.(2.1.2) with respect to x, thus,

$$\phi'(x) = \sum_{n=-\infty}^{\infty} 2a_n\phi'(2x-n) \tag{2.3.7}$$

This leads to the same eigenvalue problem as for the scaling function, only now we are looking for the eigenvector corresponding to the eigenvalue 1/2 (c.f. Eq. (2.3.3))

$$2\begin{bmatrix} a_0 & 0 & 0 & 0 \\ a_2 & a_1 & a_0 & 0 \\ 0 & a_3 & a_2 & a_1 \\ 0 & 0 & 0 & a_3 \end{bmatrix}\begin{bmatrix} \phi'(0) \\ \phi'(1) \\ \phi'(2) \\ \phi'(3) \end{bmatrix} = \begin{bmatrix} \phi'(0) \\ \phi'(1) \\ \phi'(2) \\ \phi'(3) \end{bmatrix} \tag{2.3.8}$$

The normalizing condition for the derivative arises from the reproducing property of the scaling function. That is:

$$\int_{-\infty}^{\infty} y\ \phi(x - y)\ dy = x \tag{2.3.9}$$

Differentiating Eq. (2.3.9) with respect to x, yields:

$$\int_{-\infty}^{\infty} y\, \phi'(x - y)\, dy = 1 \tag{2.3.10}$$

Thus,

$$\sum_{n=-\infty}^{\infty} x_I\, \phi'(x - x_I) = 1 \tag{2.3.11}$$

Eq. (2.3.11) gives the normalizing condition for the derivative. The derivative of D4 is shown in Figure 2.5.

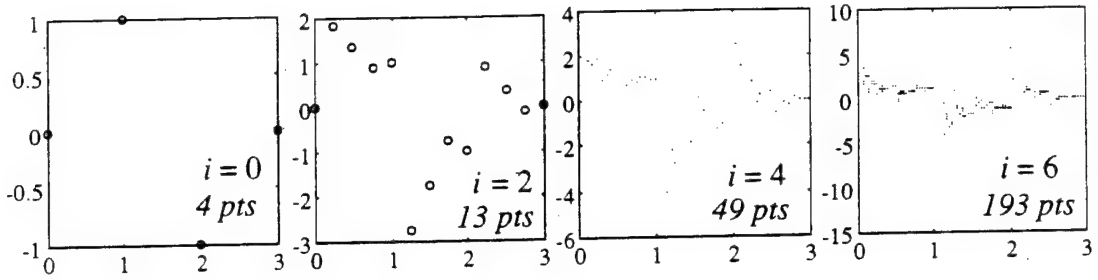


Figure 2.5 The derivative of D4 scaling function

Remark:

Scaling functions which are constructed by iterative or recursion methods are defined only at dyadic points ($2^j, j \geq 0$). Therefore, these scaling functions and their derivatives are only weakly continuous and they are often known as fractal interpolation functions. When $j \to \infty$, they converge to a continuous function.

3. FOURIER ANALYSIS OF ORTHOGONAL SCALING FUNCTIONS

3.1 Orthogonal Scaling Function

An orthogonal scaling function can also be constructed by the Fourier transform. Rewriting Eq. (2.1.2) as:

$$\phi(x) = \sum_{n} c_n\, \phi_{1.n}(x) = \sum_{n} c_n \sqrt{2}\, \phi(2x - n) \quad n \in Z \tag{3.1.1}$$

where c_n are given for each particular scaling function. The Fourier transform of $\phi(x)$ is defined as:

$$\hat{\phi}(\omega) = \int_{-\infty}^{\infty} \phi(x)\, e^{-ix\omega}\, dx \tag{3.1.2}$$

By changing variables, $y = 2x\text{-}n$ and $x = (y+n)/2$, the Fourier transform of $\phi(2x - n)$ denoted by F[ ] is:

$$F[\phi(2x - n)] = \int_{-\infty}^{\infty} \frac{1}{2}\phi(y)\, e^{-i\omega(y+n)/2}\, dy = \int_{-\infty}^{\infty} \frac{1}{2}\phi(y)\, e^{-iy\omega/2} e^{-i\omega n/2} dy = \frac{1}{2}\, e^{-in\omega/2}\, \hat{\phi}(\frac{\omega}{2}) \tag{3.1.3}$$

Equation (3.1.3) depicts the property of the Fourier transform of a "*shifted function*" $\phi(2x - n)$. Thus, the Fourier transform of Eq. (3.1.1) is:

$$\hat{\phi}(\omega) = \frac{1}{\sqrt{2}} \sum_{n} c_n\, \hat{\phi}(\frac{\omega}{2})\, e^{-in\omega/2} = m_c(\frac{\omega}{2})\, \hat{\phi}(\frac{\omega}{2}) \tag{3.1.4}$$

where

$$m_c(\omega) = \frac{1}{\sqrt{2}} \sum_{n} c_n\, e^{-in\omega} \tag{3.1.5}$$

Note that the function $m_c(\omega)$ represents the connection between two different scales of the scaling function in the Fourier space.

The Parseval identity and the orthogonality condition for the scaling function $\phi(x)$ give the Kronecker-delta property:

$$\int_{-\infty}^{\infty} \phi(x)\, \overline{\phi(x\text{-}k)}\, dx = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{\phi}(\omega)\, \overline{F[\phi(x\text{-}k)]}\, d\omega = \delta_{0k}\, ; \qquad k \in Z \tag{3.1.6}$$

where $\bar{\phi}$ denotes the complex conjugate of ϕ. Using the property of the Fourier transform of a shifted function, the left hand side of Eq. (3.1.6) becomes:

$$\int_{-\infty}^{\infty} \phi(x)\, \overline{\phi(x\text{-}k)}\, dx = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{\phi}(\omega)\, \overline{\hat{\phi}(\omega)}\, e^{i\omega k}\, d\omega$$

$$= \sum_{l} \frac{1}{2\pi}\int_{2\pi l}^{2\pi(l+1)} \hat{\phi}(\omega)\overline{\hat{\phi}(\omega)}\, e^{i\omega k} d\omega = \frac{1}{2\pi}\int_{0}^{2\pi} \left(\sum_{l} \left|\hat{\phi}(\omega+2\pi l)\right|^{2}\right) e^{i\omega k} d\omega \qquad (3.1.7)$$

In deriving Eq. (3.1.7), the integral is subdivided into intervals $[2\pi l, 2\pi(l+1)]$ for $l \in Z$. In obtaining Eq. (3.1.7), the periodicity condition $e^{ik2\pi l} = 1$ has been employed. To satisfy the condition in Eq. (3.1.6), we need

$$\sum_{l} \left|\hat{\phi}(\omega+2\pi l)\right|^{2} = 1 \qquad (3.1.8)$$

which is the orthogonality condition for a general scaling function $\phi(x)$. Since $\sum_{l} \left|\hat{\phi}(\omega+2\pi l)\right|^{2} \neq 1$ for arbitrary $\hat{\phi}$, the orthogonal scaling function $\hat{\phi}^{*}(\omega)$ is defined by:

$$\hat{\phi}^{*}(\omega) = \frac{\hat{\phi}(\omega)}{\sqrt{\sum_{l} \left|\hat{\phi}(\omega+2\pi l)\right|^{2}}} \qquad (3.1.9)$$

Note that the normalization factor in Eq. (3.1.9) can be expressed in terms of a Fourier series such that:

$$\left[\sum_{l} \left|\hat{\phi}(\omega + 2\pi l)\right|^{2}\right]^{-1/2} = \sum_{n} \alpha_{n}\, e^{-in\omega}, \;\; \alpha_{n} = \frac{1}{2\pi}\int_{0}^{2\pi} \left[\sum_{l} \left|\hat{\phi}(\omega + 2\pi l)\right|^{2}\right]^{-1/2} e^{in\omega} d\omega \qquad (3.1.10a,b)$$

Thus the orthogonal scaling function Eq. (3.1.9) becomes:

$$\hat{\phi}^{*}(\omega) = \sum_{n} \alpha_n e^{-in\omega}\, \hat{\phi}(\omega) \tag{3.1.11}$$

and since the right hand side of the above is a Fourier transform of a shifted function, Eq. (3.1.11) yields:

$$\phi^{*}(x) = \sum_{n} \alpha_n\, \phi(x-n) \tag{3.1.12}$$

3.2 Orthogonal Wavelet

In order for $\psi^*(x)$ to be an orthogonal wavelet for an orthogonal scaling function $\phi^*(x)$, the inner product of $\psi^*(x)$ and $\phi^*(x\text{-}k)$ must vanish. That is

$$\left\langle \psi^*, \phi_k^* \right\rangle = \int_{-\infty}^{\infty} \psi^*(x)\, \overline{\phi^*(x\text{-}k)}\, dx = 0 \tag{3.2.1}$$

Similar to Eq. (3.1.1), $\psi^*(x)$ can be expressed as:

$$\psi^*(x) = \sum_{n} f_n^*\, \phi^*_{1,n}(x) = \sum_{n} f_n^* \sqrt{2}\, \phi^*(2x - n) \tag{3.2.2}$$

where the f_n^* coefficients are to be determined. The Fourier transform of Eq. (3.2.2) is:

$$\hat{\psi}^*(\omega) = \frac{1}{\sqrt{2}} \sum_{n} f_n^*\, \hat{\phi}^*\left(\frac{\omega}{2}\right) e^{-in\omega/2} = m_f\left(\frac{\omega}{2}\right)\, \hat{\phi}^*\left(\frac{\omega}{2}\right) \tag{3.2.3}$$

where

$$m_f(\omega) = \frac{1}{\sqrt{2}} \sum_{n} f_n^*\, e^{-in\omega} \tag{3.2.4}$$

Using Parseval's identify and the property of the Fourier transform of a shifted function Eq. (3.1.8), the Fourier transform of Eq. (3.2.1) is given as:

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{\psi}^*(\omega)\, \overline{\hat{\phi}^*(\omega)}\, e^{ik\omega} d\omega = 0 \tag{3.2.5}$$

The domain of the above integral is subdivided into intervals $[2\pi l, 2\pi(l+1)]$ and the integrals over these intervals summed. The left hand side of Eq. (3.2.5) becomes:

$$\sum_{l} \frac{1}{2\pi} \int_{2\pi l}^{2\pi(l+1)} \hat{\psi}^{*}(\omega)\, \overline{\hat{\phi}^{*}(\omega)}\, e^{ik\omega} d\omega \qquad (3.2.6)$$

By substituting $\omega' = \omega - 2\pi l$, Eq. (3.2.6) yields:

$$\frac{1}{2\pi} \sum_{1} \int_{0}^{2\pi} \hat{\psi}^{*}(\omega'+2\pi l)\, \overline{\hat{\phi}^{*}(\omega'+2\pi l)}\, e^{ik(\omega'+2\pi l)} d\omega' \qquad (3.2.7)$$

Finally, using the periodicity of $e^{i\omega}$, i.e. $e^{ik2\pi l} = 1$ and setting $\omega = \omega'$, orthogonality gives:

$$\int_{0}^{2\pi} e^{i\omega k} \sum_{l} \hat{\psi}^{*}(\omega+2\pi l)\, \overline{\hat{\phi}^{*}(\omega+2\pi l)} d\omega = 0 \qquad (3.2.8)$$

Hence

$$\sum_{l} \hat{\psi}^{*}(\omega+2\pi l)\, \overline{\hat{\phi}^{*}(\omega+2\pi l)} = 0 \qquad (3.2.9)$$

Substituting Eq. (3.2.3) and Eq. (3.1.4) into Eq. (3.2.9) yields:

$$\sum_{l} m_f(\frac{\omega}{2}+\pi l)\, \overline{m_c(\frac{\omega}{2}+\pi l)} \left| \hat{\phi}^{*}(\frac{\omega}{2}+\pi l) \right|^{2} = 0 \qquad (3.2.10)$$

From Eqs. (3.1.5) and (3.2.4), both $m_f(\omega)$ and $m_c(\omega)$ are periodical over 2π. That is:

$$\cdots = \overline{m_c(\frac{\omega}{2} - 2\pi)} = \overline{m_c(\frac{\omega}{2})} = \overline{m_c(\frac{\omega}{2} + 2\pi)} = \cdots, \text{ etc.} \qquad (3.2.11)$$

Regrouping the sums for odd and even l, and using the orthogonality condition Eq. (3.1.10), and Eq. (3.2.11), Eq. (3.2.10) leads to:

$$m_f(\omega')\overline{m_c(\omega')} + m_f(\omega'+\pi)\overline{m_c(\omega'+\pi)} = 0 \qquad \text{where } \omega' = \frac{\omega}{2} \tag{3.2.12}$$

To solve Eq. (3.2.12), let $\omega = \omega'$ and assume $m_f(\omega) = \lambda(\omega)\,\overline{m_c(\omega+\pi)}$. Eq. (3.2.12) becomes:

$$\lambda(\omega)\,\overline{m_c(\omega)}\,\overline{m_c(\omega+\pi)} + \lambda(\omega+\pi)\overline{m_c(\omega+\pi)}\,\overline{m_c(\omega+2\pi)} = 0 \tag{3.2.13}$$

From the definition of $m_c(\omega)$, Eq. (3.1.5), we have $\overline{m_c(\omega)} = \overline{m_c(\omega+2\pi)}$ and $\overline{m_c(\omega)}\,\overline{m_c(\omega+2\pi)} \neq 0$, so that Eq. (3.2.13) can be simplified as:

$$\lambda(\omega) + \lambda(\omega+\pi) = 0 \tag{3.2.14}$$

The solution to Eq. (3.2.14) is:

$$\lambda(\omega) = e^{i\omega} \tag{3.2.15}$$

therefore

$$m_f(\omega) = e^{i\omega}\,\overline{m_c(\omega+\pi)} \tag{3.2.16}$$

Employing Eq. (3.2.16) in Eq. (3.2.3), the Fourier transform of an orthogonal wavelet is:

$$\hat{\psi}^*(\omega) = m_f(\frac{\omega}{2})\,\hat{\phi}^*(\frac{\omega}{2}) = e^{i\omega/2}\,\overline{m_c(\frac{\omega}{2}+\pi)}\,\hat{\phi}^*(\frac{\omega}{2}) \tag{3.2.17}$$

Expanding $\overline{m_c(\frac{\omega}{2}+\pi)}$ via Eq. (3.1.5) yields:

$$\hat{\psi}^*(\omega) = \frac{1}{\sqrt{2}}\sum_n c_n^*\,e^{in(\omega/2+\pi)}\,e^{i\omega/2}\,\hat{\phi}^*(\frac{\omega}{2}) = \frac{1}{\sqrt{2}}\sum_n c_n^*\,e^{in\pi}\,e^{-i\omega(-n-1)/2}\,\hat{\phi}^*(\frac{\omega}{2})$$

$$= \frac{1}{\sqrt{2}}\sum_n c_n^*\,(-1)^n\,e^{-i\omega(-n-1)/2}\,\hat{\phi}^*(\frac{\omega}{2}) = \frac{1}{\sqrt{2}}\sum_n c_{-n-1}^*\,(-1)^{-n-1}\,e^{-i\omega n/2}\,\hat{\phi}^*(\frac{\omega}{2}) \qquad (3.2{:}18)$$

With Eq. (3.1.3), the inverse Fourier transform of Eq. (3.2.18) gives:

$$\psi^*(x) = \sqrt{2} \sum_n c^*_{-n-1} (-1)^{n-1} \phi^*(2x-n) \tag{3.2.19}$$

Once the coefficients c^*_n are determined, the orthogonal wavelet $\psi^*(x)$ can also be found by Eq. (3.2.19). Equating Eq. (3.2.2) to (3.2.19) gives:

$$f^*_n = c^*_{-n-1} (-1)^{n-1}, \quad n \in Z \tag{3.2.20}$$

3.3 Derivation of Orthogonal Wavelets from a Scaling Function

In order to derive an orthogonal wavelet from a scaling function, Eq. (3.1.9) is rewritten as:

$$\hat{\phi}^*(\omega) = \frac{\hat{\phi}(\omega)}{\sqrt{\sum_l \left|\hat{\phi}(\omega+2\pi l)\right|^2}}$$

$$= m_c(\tfrac{\omega}{2}) \frac{\sqrt{\sum_l \left|\hat{\phi}(\frac{\omega}{2}+2\pi l)\right|^2}}{\sqrt{\sum_l \left|\hat{\phi}(\omega+2\pi l)\right|^2}} \frac{\hat{\phi}(\frac{\omega}{2})}{\sqrt{\sum_l \left|\hat{\phi}(\frac{\omega}{2}+2\pi l)\right|^2}} = m^*_c(\tfrac{\omega}{2})\, \hat{\phi}^*(\tfrac{\omega}{2}) \tag{3.3.1}$$

In deriving Eq. (3.3.1), the expansion Eq. (3.1.4) is used to represent $\hat{\phi}(\omega)$ and the last term in Eq. (3.3.1) can be expressed as $\hat{\phi}^*(\frac{\omega}{2})$ by using Eq. (3.1.9). Hence a new m^*_c for the orthogonal scaling function $\hat{\phi}^*(\omega)$ can be defined as:

$$m^*_c(\tfrac{\omega}{2}) = m_c(\tfrac{\omega}{2}) \frac{\sqrt{\sum_l \left|\hat{\phi}(\frac{\omega}{2}+2\pi l)\right|^2}}{\sqrt{\sum_l \left|\hat{\phi}(\omega+2\pi l)\right|^2}} \tag{3.3.2}$$

Substituting Eq. (3.1.5) into Eq. (3.3.2) yield the computational formula for $m^*_c(\frac{\omega}{2})$:

$$m_c^*(\frac{\omega}{2}) = \left(\frac{1}{\sqrt{2}}\sum_n c_n\, e^{-in\frac{\omega}{2}}\right)\frac{\sqrt{\sum_l \left|\widehat{\phi}(\frac{\omega}{2}+2\pi l)\right|^2}}{\sqrt{\sum_l \left|\widehat{\phi}(\omega+2\pi l)\right|^2}} \tag{3.3.3}$$

It is noted that for each scaling function $\phi(x)$, the coefficients c_n are given. To represent $m_c^*(\frac{\omega}{2})$ via an expression similar to Eq. (3.1.5), a new set of coefficient d_n is defined through:

$$m_c^*(\frac{\omega}{2})= \frac{1}{\sqrt{2}}\sum_n d_n\, e^{-i\omega n/2} \tag{3.3.4}$$

Once $m_c^*(\frac{\omega}{2})$ is given by Eq. (3.3.3), the Fourier coefficients d_n appearing in Eq. (3.3.4) can be computed as:

$$d_n = \frac{1}{2\pi}\int_0^{2\pi} \sqrt{2}\, m_c^*(\omega')\, e^{in\omega'}\, d\omega' = \frac{1}{2\pi}\int_0^{4\pi} \frac{1}{\sqrt{2}}\, m_c^*(\frac{\omega}{2})\, e^{in\frac{\omega}{2}}\, d\omega \tag{3.3.5}$$

With d_n determined and using the expansion Eq. (3.1.1), for a given scaling function $\phi(x)$, its orthogonal scaling function is:

$$\phi^*(x) = \sqrt{2}\sum_{\mathrm{n}} d_n\, \phi^*(2x-n) \tag{3.3.6}$$

the corresponding orthogonal wavelet is (c.f. Eq. 3.2.19):

$$\psi^*(x) = \sqrt{2}\sum_n (-1)^{n-1}\, d_{-n-1}\, \phi^*(2x-n) \tag{3.3.7}$$

Substituting Eq. (3.1.16) into Eq. (3.3.7), the relation between an orthogonal wavelet and a scaling function that satisfies Eq. (3.1.1) can be expressed as:

$$\psi^*(x) = \sqrt{2}\sum_n (-1)^{n-1}\, d_{-n-1}\sum_m \alpha_m\, \phi(2x-m-n) \tag{3.3.8}$$

3.4 Construction Procedure

The following procedures are used to derive an orthogonal scaling function ϕ^* and an orthogonal wavelet ψ^* from a scaling function ϕ.

STEP 1: Choose ϕ so that:

1. $\phi(x)$ and $\hat{\phi}(\omega)$ to have a reasonable decay and a finite support. (3.4.1a)
2. Eq. (3.1.1) is satisfied, that is c_n are given. (3.4.1b)
3. $\int \phi(x)\, dx \neq 0$ (3.4.1c)

STEP 2: Compute $\sum_l \left|\hat{\phi}(\omega+2\pi l)\right|^2$

The coefficient, $\sum_l \left|\hat{\phi}(\omega+2\pi l)\right|^2$, can be expressed in terms of a Fourier series such that:

$$\sum_l \left|\hat{\phi}(\omega+2\pi l)\right|^2 = \sum_n b_n e^{-i\omega n}\,; \qquad b_n = \frac{1}{2\pi}\int_0^{2\pi} \sum_l \left|\hat{\phi}(\omega+2\pi l)\right|^2 e^{i\omega n}\, d\omega \qquad (3.4.2a,b)$$

Note that the expression in Eq. (3.4.2a,b) is essentially the same as the last expression in Eq. (3.1.7). Therefore, Eq. (3.4.2a,b) is simplified:

$$b_n = \int_{-\infty}^{\infty} \phi(x)\, \overline{\phi(x-k)}\, dx \qquad (3.4.3)$$

In general, at least $(m+1)/2$ Gauss points are needed for the numerical integration of Eq. (3.4.3), where m is the highest order of the product $\phi(x)\ \phi(x-k)$ in Eq. (3.4.3).

STEP 3: Compute $\dfrac{1}{\sqrt{\sum_l \left|\hat{\phi}(\omega+2\pi l)\right|^2}}$ and coefficients α_n

Recall from Eq. (3.1.10a,b)

$$\frac{1}{\sqrt{\sum_l \left|\hat{\phi}(\omega+2\pi l)\right|^2}} = \sum_n \alpha_n e^{-in\omega}; \qquad \alpha_n = \frac{1}{2\pi}\int_0^{2\pi} \frac{1}{\sqrt{\sum_l \left|\hat{\phi}(\omega+2\pi l)\right|^2}}\, e^{in\omega} d\omega \qquad (3.4.4a,b)$$

It is noted that $\sqrt{\sum_{l} \left|\hat{\phi}(\omega+2\pi l)\right|^{2}}$ has been expressed in terms of b_n via Eq. (3.4.2a,b).

STEP 4: Construction of the orthogonal scaling function $\phi^{*}(x)$

From Eq. (3.1.12):

$$\phi^{*}(x) = \sum_{n} \alpha_n \, \phi(x-n) \tag{3.4.5}$$

The α_n coefficients are in general nonzero in Eq. (3.1.10b). In order to keep only a few terms α_n, we choose rapidly decaying scaling functions so that only a finite number of terms is necessary. In other words α_n decays rapidly away from $n = 0$. Thus, we can neglect the high order coefficients α_n.

STEP 5: Compute $m_c(\frac{\omega}{2})$

We employ the scaling function $\phi(x)$ that satisfies Eq. (3.1.1) with c_n given. From Eq. (3.1.5):

$$m_c(\frac{\omega}{2}) = \frac{1}{\sqrt{2}} \sum_{n} c_n \, e^{-in\frac{\omega}{2}} \tag{3.4.6}$$

STEP 6: Compute $m_c^{*}(\frac{\omega}{2})$ and the wavelet coefficients d_n

From Eq. (3.3.2)

$$m_c^{*}(\frac{\omega}{2}) = m_c(\frac{\omega}{2}) \frac{\sqrt{\sum_{l} \left|\hat{\phi}(\frac{\omega}{2}+2\pi l)\right|^{2}}}{\sqrt{\sum_{l} \left|\hat{\phi}(\omega+2\pi l)\right|^{2}}} = \frac{1}{\sqrt{2}} \sum_{n} d_n \, e^{-in\frac{\omega}{2}} \tag{3.4.7}$$

where d_n are the Fourier coefficients. They can be obtained via a Fourier series expansion:

$$d_n = \frac{1}{2\pi} \int_{0}^{4\pi} \frac{1}{\sqrt{2}} m_c^{*}(\frac{\omega}{2}) \, e^{in\omega/2} d\omega \tag{3.4.8}$$

STEP 7: Construct the orthogonal wavelet $\psi^{*}(x)$

From Eq. (3.3.8)

$$\psi^*(x) = \sqrt{2} \sum_{-n-1} (-1)^{n-1} d_{-n-1} \sum_{m} \alpha_m \, \phi(2x-m-n) \tag{3.4.9}$$

Remark: Recall from Eq. (3.3.6):

$$\phi^*(x) = \sqrt{2} \sum_{\mathrm{n}} d_n \, \phi^*(2x-n) \tag{3.4.10}$$

Substituting Eq. (3.1.12) into Eq. (3.4.10) yields an alternative way to construct the orthogonal scaling function:

$$\phi^*(x) = \sqrt{2} \sum_{\mathrm{n}} d_n \left[\sum_{m} \alpha_m \, \phi(2x-m-n) \right] \tag{3.4.11}$$

It is noted that the construction of an orthogonal scaling function $\phi^*(x)$ by using Eq. (3.4.11) should be identical to that of using Eq. (3.4.5).

STEP 8: Fourier Transform of $\phi^*(x)$ and $\psi^*(x)$

From Eq. (3.1.9), the Fourier transform of the orthogonal scaling function is given by:

$$\hat{\phi}^*(\omega) = \frac{\hat{\phi}(\omega)}{\sqrt{\sum_{l} \left|\hat{\phi}(\omega+2\pi l)\right|^2}} = \left(\sum_{n} \alpha_n \, e^{-in\omega} \right) \hat{\phi}(\omega) \tag{3.4.12}$$

with α_n defined in Eq. (3.1.10b). Similarly for Eq. (3.2.18), the Fourier transform of the orthogonal wavelet is:

$$\hat{\psi}^*(\omega) = \frac{1}{\sqrt{2}} \sum_{n} d_{-n-1} \, (-1)^{-n-1} \, e^{-i\omega n/2} \, \hat{\phi}^*\left(\frac{\omega}{2}\right) \tag{3.4.13}$$

Example 1. The Piecewise Linear Spline Function

To provide a simple example, we choose ϕ to be the piecewise linear spline function,

$$\phi(x) = \begin{cases} 1+x & -1 \le x \le 0 \\ 1-x & 0 < x \le 1 \\ 0 & \text{otherwise} \end{cases}$$

as plotted in Figure 3.1a.

STEP 1: This function $\phi(x)$ satisfies [see Fig. 3.1a; Eq.(3.1.1)]

$$\phi(x) = \frac{1}{2}\phi(2x-1) + \phi(2x) + \frac{1}{2}\phi(2x+1);\ c_{-1} = \frac{1}{2\sqrt{2}},\ c_0 = \frac{1}{\sqrt{2}},\ c_1 = \frac{1}{2\sqrt{2}}$$

STEP 2: From Eq. (3.4.3), b_n are computed as (two-point Gaussian integration is used):

$$b_{-1} = \frac{1}{6},\ b_0 = \frac{2}{3},\ b_1 = \frac{1}{6}$$

Thus the normalization coefficient, Eq. (3.4.2a), is given as:

$$\sum_l \left|\hat{\phi}(\omega+2\pi l)\right|^2 = \sum_n b_n e^{-i\omega n} = \frac{1}{6}e^{-i\omega} + \frac{2}{3}e^0 + \frac{1}{6}e^{i\omega} = \frac{2}{3} + \frac{1}{6}\cos\omega$$

STEP 3: The orthogonal expansion coefficients α_n as defined in Eqs. (3.4.4a,b) are tabulated in Table 3.1.

n	α_n
0	1.29167549215928
1	-0.174663234444464
2	3.521011525921824E-02
3	-7.874425129503932E-03

Table 3.1 Orthogonal expansion coefficients

Note that $\alpha_n = \alpha_{-n}$, and the high order coefficients α_n are neglected.

STEP 4: The orthogonal scaling function can be computed via Eq.(3.4.5). It is shown in Figure 3.1b.

STEP 5: The computation of $m_c(\frac{\omega}{2})$. From Eq. (3.4.6)

$$m_c(\frac{\omega}{2}) = \frac{1}{\sqrt{2}} \sum_n c_n\, e^{-in\frac{\omega}{2}} = \frac{1}{\sqrt{2}} (\frac{1}{2\sqrt{2}} e^{i\frac{\omega}{2}} + \frac{1}{\sqrt{2}} + \frac{1}{2\sqrt{2}} e^{-i\frac{\omega}{2}}) = \frac{1}{2} (1 + \cos \frac{\omega}{2})$$

STEP 6: The computation of $m_c^*(\frac{\omega}{2})$. From Eq. (3.4.7)

$$m_c^*(\frac{\omega}{2}) = m_c(\frac{\omega}{2}) \frac{\sqrt{\sum_l \left|\hat{\phi}(\frac{\omega}{2}+2\pi l)\right|^2}}{\sqrt{\sum_l \left|\hat{\phi}(\omega+2\pi l)\right|^2}} = \frac{1}{2} (1 + \cos \frac{\omega}{2}) \frac{\sqrt{\frac{2}{3} + \frac{1}{6} \cos \frac{\omega}{2}}}{\sqrt{\frac{2}{3} + \frac{1}{6} \cos \omega}}$$

The wavelet coefficients d_n can be computed according to Eq. (3.4.8) and they are given in Table 3.2.

n	d_n
0	0.8176464057010934
1	0.3972970881341911
2	-6.910098674164672E-02
3	-5.194534808183847E-02
4	1.697104789387069E-02
5	9.990595444183770E-03
6	-3.883262250905551E-03
7	-2.201951238397208E-03
8	9.233710054871564E-04
9	5.116360226930585E-04
10	-2.242963267262629E-04

Table 3.2 Wavelet coefficients, d_n for linear spline function

Note that $d_n = d_{-n}$. The coefficients d_n are nonzero for all n and the higher order coefficients are neglected.

STEP 7: Using expression Eqs. (3.4.11), the orthogonal wavelet is shown in Figures 3.1c.

STEP 8: The Fourier transform of $\phi^*(x)$ and $\psi^*(x)$. For the linear spline function $\phi(x)$, the Fourier transform is given by (see Figure 3.1d):

$$\hat{\phi}(\omega) = \left(\frac{\sin \frac{\omega}{2}}{\frac{\omega}{2}} \right)^2$$

Thus the Fourier transforms of $\hat{\phi}^*(x)$ and $\hat{\psi}^*(x)$ are given by (see Fig.3.1e and 3.1f, respectively):

$$\hat{\phi}^*(\omega) = \left(\frac{\sin \frac{\omega}{2}}{\frac{\omega}{2}} \right)^2 \frac{1}{\sqrt{\frac{2}{3} + \frac{1}{6} \cos \omega}}; \quad \hat{\psi}^*(\omega) = \frac{1}{\sqrt{2}} \sum_n d_{-n-1} \, (-1)^{-n-1} \, e^{-i\omega n/2} \left(\frac{\sin \frac{\omega}{4}}{\frac{\omega}{4}} \right)^2 \frac{1}{\sqrt{\frac{2}{3} + \frac{1}{6} \cos \frac{\omega}{2}}}$$

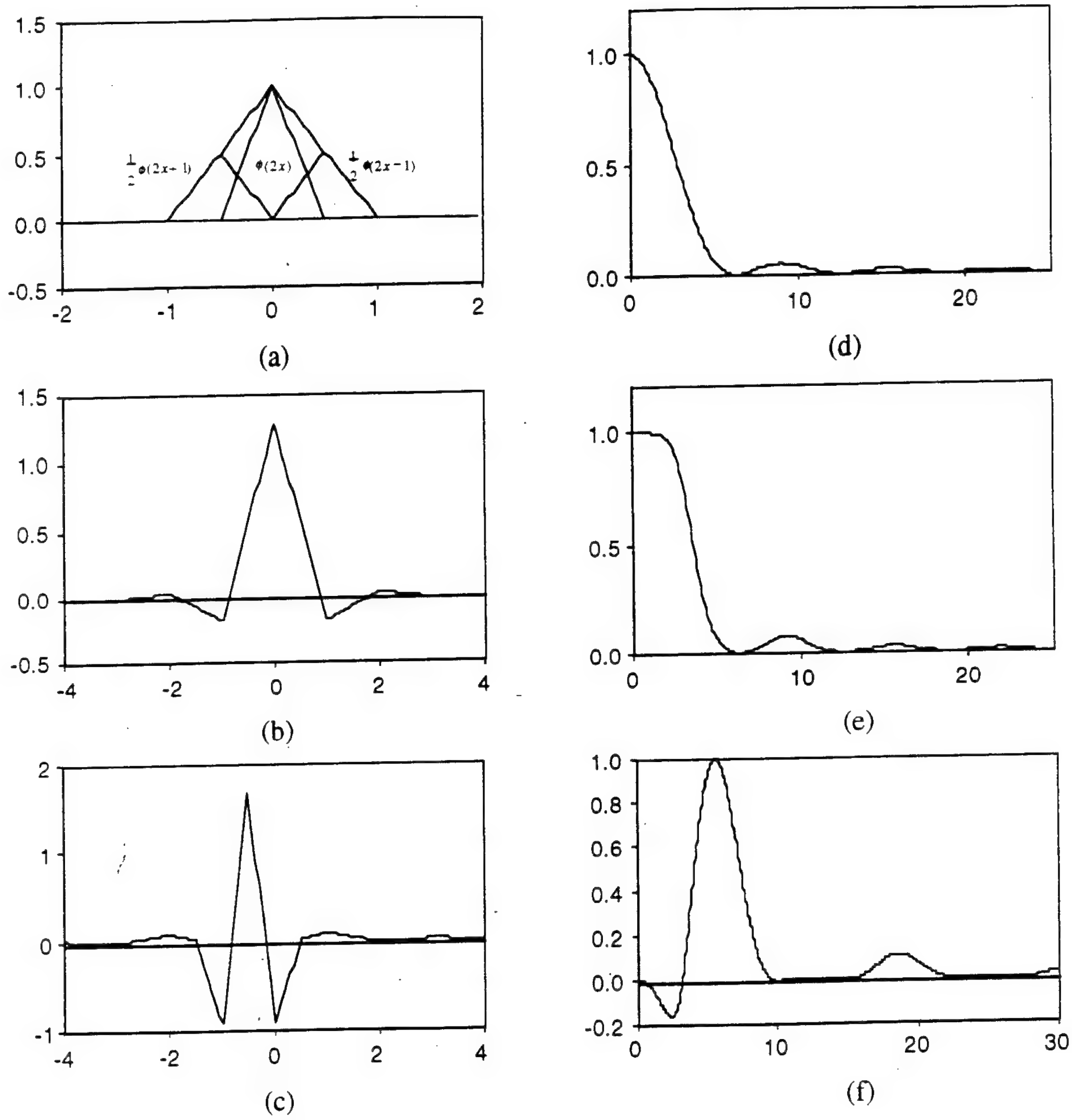


Figure 3.1 (a) The piecewise linear spline function, (b) the corresponding orthogonal scaling function , (c) the wavelet, (d) the Fourier transform of the linear spline function , (e) orthogonal scaling function, (f) orthogonal wavelet.

Example 2. High Order Spline Functions

In this example, we present the various coefficients generated for the cubic spline and 5th order spline function (Tables 3.3 - 3.7). The spatial and Fourier transform space for the scaling function and wavelet are depicted in Figures 3.2 and 3.3.

Comparing the frequency spectrum of the linear, cubic and 5th order orthogonal scaling functions and wavelets (Figs. 3.1e and 3.1f, Figs. 3.1e and 3.1f and Figs. 3.1e and 3.1f, respectively), it can be seen that the higher order scaling function approximates a rectangular low-pass filter, whereas the higher order wavelets have better high-pass filter properties. It is noted that, the "side-lobes" are decreased when the order of the spline increases.

$\phi(x)$ (cubic spline function)	$\phi(x)$ (5th order spline function)
$\frac{1}{6}(x+2)^3, \quad -2 \le x \le -1$	$\frac{1}{120}(3+x)^5, \quad -3 \le x \le -2$
$\frac{2}{3} - x^2(1+\frac{x}{2}), \quad -1 \le x \le 0$	$\frac{1}{120}(51 - 75x - 210x^2 - 150x^3 - 45x^4 - 5x^5), \quad -2 \le x \le -1$
$\frac{2}{3} - x^2(1-\frac{x}{2}), \quad 0 \le x \le 1$	$\frac{1}{60}(33 - 30x^2 + 15x^4 + 5x^5), \quad -1 \le x \le 0$
$\frac{-1}{6}(x-2)^3, \quad 1 \le x \le 2$	$\frac{1}{60}(33 - 30x^2 + 15x^4 - 5x^5), \quad 0 \le x \le 1$
	$\frac{1}{120}(51 + 75x - 210x^2 + 150x^3 - 45x^4 + 5x^5), \quad 1 \le x \le 2$
	$\frac{1}{120}(3-x)^5, \quad 2 \le x \le 3$

Table 3.3 The cubic and 5th order spline functions

The coefficients c_n :

n	c_n (cubic spline function)	c_n (5th order spline function)
-3	0	$\frac{1}{32\sqrt{2}}$
-2	$\frac{1}{8\sqrt{2}}$	$\frac{3}{16\sqrt{2}}$
-1	$\frac{1}{2\sqrt{2}}$	$\frac{15}{32\sqrt{2}}$
0	$\frac{3}{4\sqrt{2}}$	$\frac{5}{8\sqrt{2}}$
1	$\frac{1}{2\sqrt{2}}$	$\frac{15}{32\sqrt{2}}$
2	$\frac{1}{8\sqrt{2}}$	$\frac{3}{16\sqrt{2}}$
3	0	$\frac{1}{32\sqrt{2}}$

Table 3.4 Coefficients c_n

The coefficients b_n :

n	b_n (cubic spline function)	b_n (5th order spline function)
-4	0	0.393925565271235
-3	1.984126820776084E-04	0.243960287323605
-2	2.380952406175427E-02	5.520202023558386E-02
-1	0.2363095229476177	3.823878659712875E-03
0	0.4793650806171007	5.100609337337319E-05
1	0.2363095229476177	3.823878659712875E-03
2	2.380952406175427E-02	5.520202023558386E-02
3	1.984126820776084E-04	0.243960287323605
4	0	0.393925565271235

Table 3.5 Coefficients b_n

The orthogonal expansion coefficients α_n :

n	α_n (cubic spline function)	α_n (5th order spline function)
0	1.96976165984483	3.21252767040872
1	-.672430465527351	-1.67129203556098
2	.2687042287778821	.8693729316036274
3	-.118519934970734	-.476253034889491
4	5.519145833683330E-02	.2724049771107151
5	-2.652033542024175E-02	-.160669074515415
6	1.299816380236143E-02	9.681114156185701E-02
7	-6.457490594044601E-03	-5.921982820667219E-02
8	3.239862395307044E-03	3.662439553866464E-02
9	-1.637775639515686E-03	-2.283772914274113E-02
10	8.328359233600000E-04	1.433219729959393E-02

Table 3.6 Orthogonal expansion coefficients

The wavelet coefficients d_n :

n	d_n (cubic spline function)	d_n (5th order spline function)
0	0.7661300483768689	.7472336463549821
1	0.4339226328986687	.442463060087634
2	-5.020171992889517E-02	-3.701993636800858E-02
3	-.11003701684689	-.129268688614172
4	3.208089410823325E-02	2.947409035770669E-02
5	4.206834992601178E-02	6.131245571925825E-02
6	-1.717631339398893E-02	-2.100611348868239E-02
7	-1.798231987132275E-02	-3.252003030074334E-02
8	8.685293562579883E-03	1.400951847251998E-02
9	8.201476481521336E-03	1.820860315855871E-02
10	-4.353838725200284E-03	-9.049154645280076E-03

Table 3.7 Wavelet coefficients

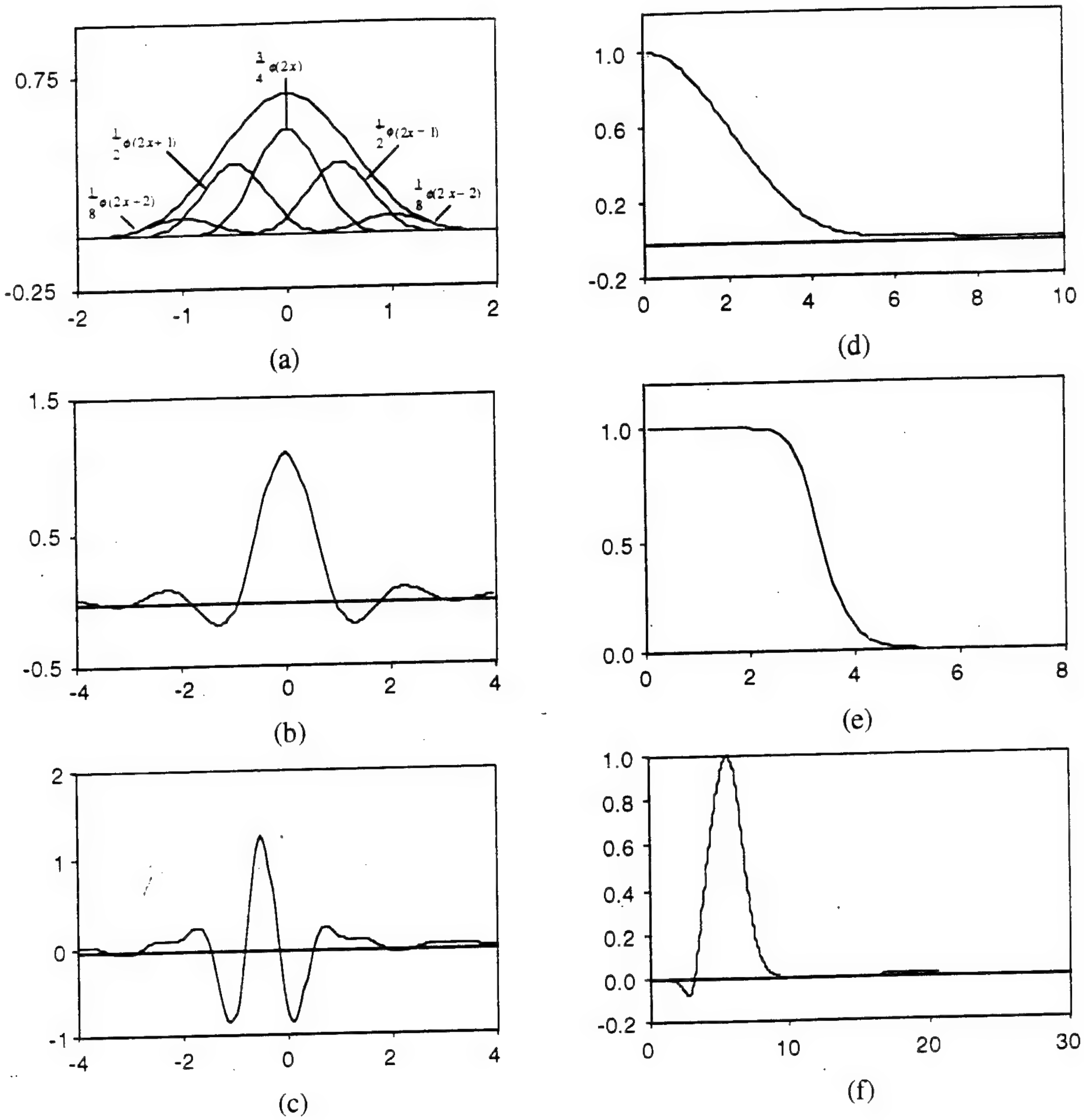
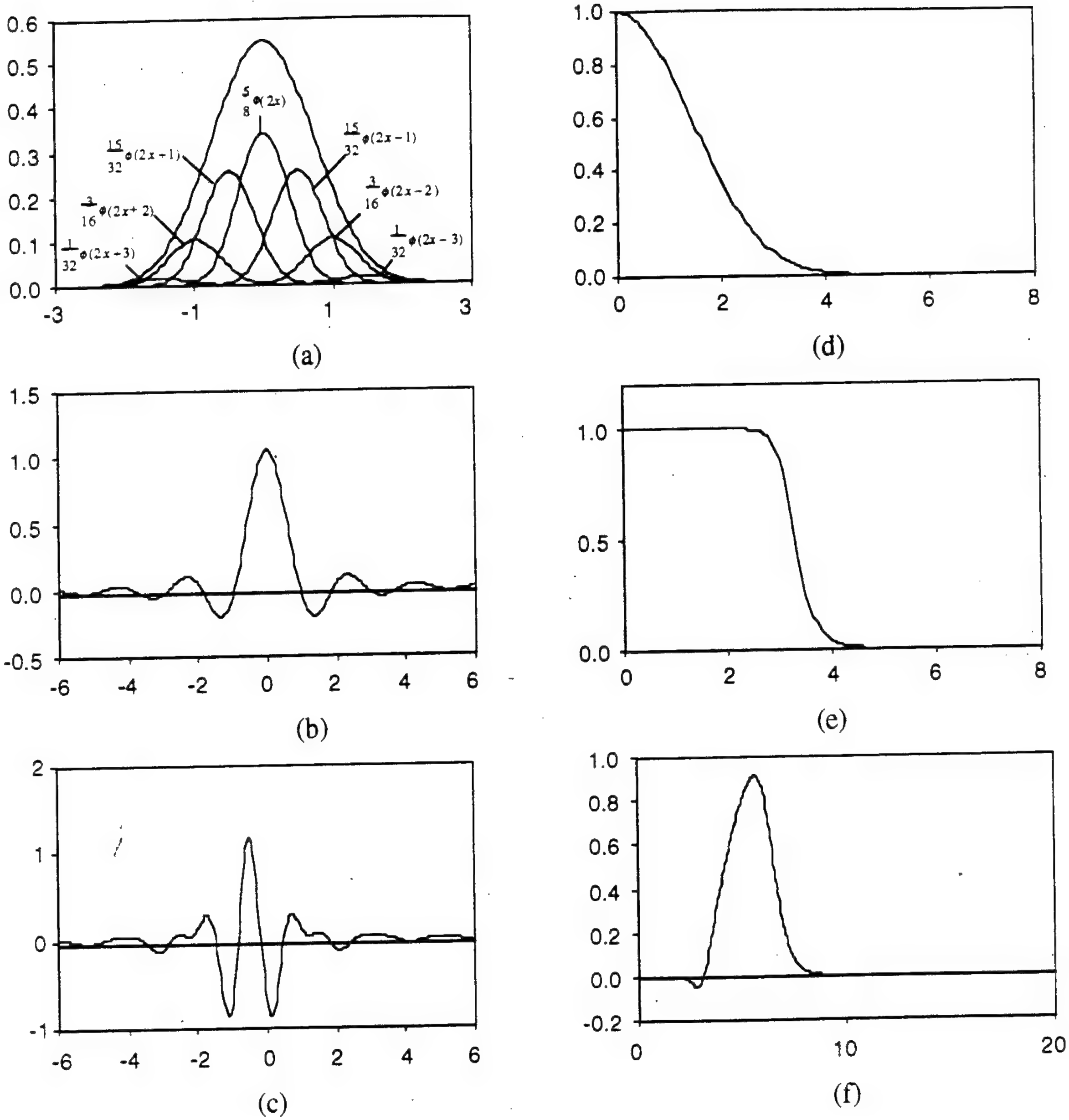


Figure 3.2 (a) The cubic spline function, (b) the orthogonal scaling function, (c) the wavelet, (d) the Fourier transform of cubic spline function, (e) orthogonal scaling function, (f) orthogonal wavelet (e)



(a)

(d)

(b)

(e)

(c)

(f)

Figure 3.3 (a) The 5th order spline function, (b) the orthogonal scaling function, (c) the wavelet, (d) the Fourier transform of 5th order spline function, (e) orthogonal scaling function, (f) orthogonal wavelet.

4. REPRODUCING KERNEL METHODS (RKM)

4.1 Relations between Convolution and Reproducing Kernel Methods

Given a sufficiently smooth function *u(x)*, we wish to construct a window function $\phi(x) \in C^{n+1}$, which is also called a reproducing kernel, such that

$$u^R(x) = \int_{-\infty}^{\infty} \phi(x - y)\, u(y)\, dy \tag{4.1.1}$$

For computational efficiency, $\phi(x)$ is chosen so that it is non-zero only in a compact support, i.e. for every *B(y)*,

$$\phi(x - y) = \begin{cases} > 0 & x \in B(y) \\ 0 & x \notin B(y) \end{cases}$$

Equation (4.1.1) is can also be expressed as a convolution integral, ie, $\phi(x) * u(x) = u^R(x)$. From the convolution theorem, the spatial convolution $u^R(x)$ is equivalent to a multiplication in the Fourier domain.

$$\widehat{u}^R(\xi) = \widehat{u}(\xi)\, \widehat{\phi}(\xi) \tag{4.1.2}$$

The physical meaning of the convolution Eq. (4.1.1) can not be explained clearly either verbally or pictorially. However, it becomes more transparent when we study its Fourier transform counterpart.

In order to control the frequency/wave number content or the 'scale', we introduce a scaling parameter (dilation parameter or refinement parameter are other names) $a > 0$ so that Eq. (4.1.1) becomes

$$u^{R_a}(x) \equiv P_a u(x) \equiv \int_{-\infty}^{\infty} \phi_a(x - y) u(y) dy \tag{4.1.3}$$

where the projection operator, P_a, is defined as:

$$P_a u(x) \equiv \int_{-\infty}^{\infty} \phi_a(x-y)u(y)dy \tag{4.1.4}$$

and $a(y) \equiv r * h(y)$ with $r = 2^j$ $(j = -\infty,\cdots,-2,-1,0,1,2,\cdots,\infty)$ and $h(y)$ is the nodal spacing function. Based on the resolution of the projection operator, $P_{a/2^n}$, a hierarchical representation of the function $u(x)$ is defined as

$$u(x) = \lim_{j\to-\infty} P_{2^j a}u(x) \supset \ldots P_a u(x) \supset P_{2a}u(x) \supset P_{4a}u(x) \supset \ldots \supset \lim_{j\to\infty} P_{2^j a}u(x) = \{0\} \tag{4.1.5}$$

Now, we introduce a wavelet projection given by

$$Q_{2a}\, u(x) \equiv \int_{-\infty}^{\infty} \psi_{2a}(x - y)\, u(y)\, dy \quad . \tag{4.1.6}$$

where $\psi_{2a}(x - y)$ is the $2a$ scale wavelet. It is defined by

$$\psi_{2a}(x - y) = \phi_a(x - y) - \phi_{2a}(x - y) \tag{4.1.7}$$

Rewriting Eq. (4.1.4) together with Eq. (4.1.7) yields

$$P_a u(x) = \int_{-\infty}^{\infty} \left[\phi_a(x - y) - \phi_{2a}(x - y) + \phi_{2a}(x - y)\right]u(y)\, dy$$

$$= \int_{-\infty}^{\infty} \phi_{2a}(x - y)u(y)\, dy + \int_{-\infty}^{\infty} \psi_{2a}(x - y)u(y)\, dy$$

$$= P_{2a}u(x) + Q_{2a}u(x) = P_{4a}u(x) + Q_{4a}u(x) + Q_{2a}u(x) = \ldots \text{etc.} \tag{4.1.8}$$

Equation (4.1.8) illustrates the framework of multiresolution analysis, which can be described by sequences of nested closed subspaces defined in Eq. (4.1.5). If a complementary projection operator, Q_{2a}, is defined as in Eq. (4.1.6), then the higher scale projected solution, $P_a u$, can be represented by the sum of the next lower-level scaling projection and wavelet projection. The Q_{2a} projection can be viewed as a "peeled off" scale or the "rate of variation" of $P_a u$ The wavelet projection Eq. (4.1.6) together with the recursive two-level decomposition Eq. (4.1.8) constitute the backbone of the wavelet RKM.

4.2 Reproducing Kernel Method In One Dimension

Reproducing Condition

Consider the reproducing equation as:

$$u^{R_a}(x) = \int_{-\infty}^{\infty} u(y)\phi_a(x-y)dy \tag{4.2.1}$$

The Taylor series expansion for $u(y)$ about x is:

$$u(y) = u(x) - (x-y)u^{'}(x) + \frac{1}{2!}(x-y)^2 u^{''}(x) + \ldots$$

$$+\frac{(-1)^n}{n!}(x-y)^n u^{(n)}(x) + \frac{(-1)^{n+1}}{(n+1)!}(x-y)^{n+1}u^{(n+1)}(\bar{x}) \tag{4.2.2}$$

where $\bar{x} = x + \theta(y-x)$ and $0 < \theta \leq 1$. Substitute Eq. (4.2.2) into Eq. (4.2.1), which yields

$$u^{R_a}(x) = u(x)m_0(a,x) - u^{'}(x)m_1(a,x) + \frac{u^{''}}{2!}(x)m_2(a,x) + \ldots.$$

$$+\frac{(-1)^n}{n!}u^{(n)}(x)m_n(a,x) + \frac{(-1)^{n+1}}{n!}u^{(n+1)}(\bar{x})m_{n+1}(a,x) \tag{4.2.3}$$

where the kth moment of the window function is defined as:

$$m_k(a.x) = \int_{-\infty}^{+\infty} (x-y)^k \phi_a(x-y)dy \qquad k = 0,\ 1,\ \ldots,\ n \tag{4.2.4}$$

Apparently, if we want to reproduce $u^{R_a}(x)$ correctly up to nth order, the following reproducing conditions need to be satisfied

$$m_\alpha(a,x) = \delta_{0\alpha} \qquad \alpha = 0,\ 1,\ 2,\ ...,\ n \tag{4.2.5}$$

An arbitrary choice of a window function, however, can not guarantee the satisfaction of all the reproducing conditions Eq. (4.2.5). Therefore, we define a modified window function so that all the reproducing conditions can be satisfied. In general, this modified window function can be expressed as:

$$\overline{\phi}_a(x;x-y) = \sum_{k=0}^{n} b_k(a,x)(x-y)^k \phi_a(x-y) = \boldsymbol{P}^T(x-y)\boldsymbol{b}(a,x)\phi_a(x-y)$$

$$= C_a(x;x-y)\phi_a(x-y) \tag{4.2.6}$$

where

$$\boldsymbol{P}^T(x-y) = \left[1,(x-y),.....,(x-y)^n\right] \tag{4.2.6a}$$

and

$$\boldsymbol{b}^T(a,x) = \left[b_0(a,x),b_1(a,x),...,b_n(a,x)\right] \tag{4.2.6b}$$

Note that a polynomial type vector is adopted to the basis function $\boldsymbol{P}^T$ in Eq. (4.2.6a). However, it is not the only option for the basis function in RKM. In general, the basis function can be extended to any independent functions. The product of $\boldsymbol{P}^T(x-y)$ and $\boldsymbol{b}(a,x)$, $C_a(x;x-y)$, is called the correction function. The coefficients $b_k(a,x)$ are solved from the reproducing conditions. By this definition, the moments of the modified window function are written as

$$\overline{m}_k(a,x) = \int_{-\infty}^{+\infty} (x-y)^k \overline{\phi}_a(x;x-y)dy \qquad k = 0,\ 1,\ 2...,\ n$$

$$= \int_{-\infty}^{+\infty} (x-y)^k \left[\sum_{j=0}^{n} b_j(a,x)(x-y)^j \phi_a(x-y) \right] dy$$

$$= b_0(a,x)m_k(a,x) + b_1(a,x)m_{k+1}(a,x) + \ldots + b_n(a,x)m_{k+n}(a,x) \qquad (4.2.7)$$

or in matrix form

$$\overline{\boldsymbol{m}}(a,x) = \boldsymbol{M}(a,x)\boldsymbol{b}(a,x) \qquad (4.2.8)$$

where $\boldsymbol{M}(a, x)$ is the moment matrix:

$$\boldsymbol{M}(a,x) = \begin{bmatrix} m_0(a,x) & m_1(a,x) & \cdots & m_n(a,x) \\ m_1(a,x) & m_2(a,x) & \cdots & m_{n+1}(a,x) \\ \vdots & & & \vdots \\ m_n(a,x) & m_{n+1}(a,x) & \cdots & m_{2n}(a,x) \end{bmatrix} \qquad (4.2.9)$$

To solve the coefficients $b_k(a,x)$, we apply the reproducing conditions Eq. (4.2.5) to the modified window function. That is,

$$[\overline{m}_0(a,x), \overline{m}_1(a,x), \cdots, \overline{m}_n(a,x)]^T = [1,0,\cdots,0]^T = \boldsymbol{P}^T(0) \qquad (4.2.10)$$

Equations (4.2.8) and (4.2.10) give:

$$\boldsymbol{M}(a,x)\boldsymbol{b}(a,x) = \boldsymbol{P}(0) \qquad (4.2.11)$$

Then, the coefficient vector $\boldsymbol{b}(a,x)$ can be obtained by:

$$\boldsymbol{b}(a,x) = \boldsymbol{M}^{-1}(a,x)\boldsymbol{P}(0) \qquad (4.2.12)$$

Reproducing Conditions for the First Derivative

The first derivative of the reproducing equation Eq. (4.2.1) is defined as:

$$\frac{d}{dx}u^{R_a}(x) = \int_{-\infty}^{\infty} u(y)\frac{d}{dx}\phi_a(x-y)dy = \int_{-\infty}^{\infty} u(y)D^1\phi_a(x-y)dy \qquad (4.2.13)$$

Substituting Eq. (4.2.2) into Eq. (4.2.13) yields

$$\frac{d}{dx}u^{R_a}(x) = u(x)m_{0.1}(a,x) - u^{'}(x)m_{1.1}(a,x)$$

$$+\frac{u^{''}(x)}{2!}m_{2.1}(a,x)+\ldots+\frac{(-1)^n}{n!}u^{(n)}(x)m_{n.1}(a,x)$$

$$+\frac{(-1)^{n+1}}{(n+1)!}u^{(n+1)}(\overline{x})m_{n+1.1}(a,x) \qquad (4.2.14)$$

where $m_{\alpha.1}(a,x)$ is defined as:

$$m_{\alpha.1}(a,x) = \int_{-\infty}^{\infty}(x-y)^{\alpha}D^{1}\phi_{a}(x-y)dy \qquad \alpha = 0, 1, 2, \ldots, n \qquad (4.2.15)$$

By examining Eq. (4.2.14), the reproducing conditions up to nth order for the first derivative are:

$$m_{\alpha.1}(a,x) = -\delta_{\alpha 1} \qquad \alpha = 0, 1, 2, \ldots, n \qquad (4.2.16)$$

In general, if we define the γth derivative of the reproducing equation as:

$$D^{\gamma}u^{R_a}(x) = \int_{-\infty}^{\infty}u(y)D^{\gamma}\phi_{a}(x-y)dy \qquad (4.2.17)$$

where 'D' denotes a differential operator, then, the reproducing conditions for the γth derivative can also be obtained:

$$m_{\alpha.\gamma}(a,x) = (-1)^{\alpha}\alpha!\delta_{\alpha\gamma} \qquad (4.2.18)$$

where

$$m_{\alpha.\gamma}(a,x) = \int_{-\infty}^{\infty}(x-y)^{\alpha}D^{\gamma}\phi_{a}(x-y)dy \qquad (4.2.19)$$

Example 1: From Eq. (4.2.6), to reproduce the first derivative with up to second order accuracy, the modified window function is obtained in the form of:

$$\overline{\phi}_a(x;x-y)=\left[b_0(a,x)+b_1(a,x)(x-y)+b_2(a,x)(x-y)^2\right]\phi_a(x-y) \tag{4.2.20}$$

Applying the modified window function to the reproducing condition for the derivative and by introducing the reproduction condition $\overline{m}_\alpha(a,x)=\delta_{0\alpha}$ yields:

$$\begin{bmatrix} \dot{m}_0 & \dot{m}_1 & \dot{m}_2 & m_0 & m_1 & m_2 \\ \dot{m}_1 & \dot{m}_2 & \dot{m}_3 & m_1 & m_2 & m_3 \\ \dot{m}_2 & \dot{m}_3 & \dot{m}_4 & m_2 & m_3 & m_4 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \dot{b}_0 \\ \dot{b}_1 \\ \dot{b}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.2.21}$$

Combining Eq. (4.2.11) and Eq. (4.2.21) gives

$$\begin{bmatrix} \boldsymbol{M}(a,x) & \boldsymbol{0} \\ \dot{\boldsymbol{M}}(a,x) & \boldsymbol{M}(a,x) \end{bmatrix} \begin{bmatrix} \boldsymbol{b}(a,x) \\ \dot{\boldsymbol{b}}(a,x) \end{bmatrix} = \begin{bmatrix} \boldsymbol{P}(0) \\ \boldsymbol{0} \end{bmatrix} \tag{4.2.22}$$

Taking the first derivative on both sides of Eq. (4.2.11), yields:

$$\dot{\boldsymbol{M}}(a,x)\boldsymbol{b}(a,x)+\boldsymbol{M}(a,x)\dot{\boldsymbol{b}}(a,x)=\boldsymbol{0} \tag{4.2.23}$$

This shows that the reproducing conditions for the first derivative can also be achieved by taking the derivative of the reproducing conditions directly. That is:

$$\overline{m}_{\alpha.1}(a,x)=-\delta_{\alpha 1} \qquad \Leftrightarrow \qquad D^1\left[\boldsymbol{M}(a,x)\boldsymbol{b}(a,x)\right]=D^1\boldsymbol{P}(0) \tag{4.2.24}$$

Recalling the reproducing conditions for the γth derivative is given as:

$$\overline{m}_{\alpha.\gamma}(a,x)=(-1)^\alpha\alpha!\delta_{\alpha\gamma} \tag{4.2.25}$$

It can be shown that these conditions simply imply:

$$D^{\gamma}\overline{\mathbf{m}}(a,x) = D^{\gamma}\left[\mathbf{M}(a,x)\mathbf{b}(a,x)\right] = \mathbf{0} \tag{4.2.26}$$

4.3 Orthogonal Conditions for RKM

We define a new correction function associated with the reproducing and orthogonality conditions for a discrete system with uniform spacing, as:

$$C_a(x,x_J) = \left[\beta_0 + \beta_1(\frac{x-x_J}{a}) + \beta_2(\frac{x-x_J}{a})^2 + \ldots + \beta_N(\frac{x-x_J}{a})^N\right] \tag{4.3.1}$$

thus

$$\phi^*(\frac{x-x_J}{a}) = \sum_{\alpha=0}^{N=n+m} \beta_\alpha(\frac{x-x_J}{a})^\alpha \phi(\frac{x-x_J}{a}) \tag{4.3.2}$$

where 'n' is the order of accuracy of the reproducing conditions and 'm' is the number of nodes covered by the support of the window function $\phi(\frac{x}{a})$. Hence, a new set of coefficients β_α, need to be determined from both the reproducing and orthogonality conditions.

Reproducing Conditions

From Eq. (4.2.5), for $\phi_a^*(\frac{x-x_J}{a})$ to reproduce the solution up to nth order requires:

$$\sum_{i=0}^{n+m} m_{i+j}\beta_i = \delta_{0j} \qquad j = 0,\ 1,\ \ldots,\ n \tag{4.3.3}$$

There are $n+1$ equations with $n+m+1$ unknowns. In order to obtain a complete set of β_α (α = 0, 1, ..., n, $n+1$, ..., $n+m$), another m equations can be acquired from the orthogonality conditions.

Orthogonality Conditions for Scaling Function

Instead of solving for the filter coefficients a_n, the same orthogonality conditions can be applied to solve the correction coefficients β_α. Furthermore, with the property of compact support for the window function, only the nodes covered by the support need to be considered in the orthogonality condition. That is:

$$\int_{-\infty}^{\infty} \phi^*(x)\phi^*(x-k) = \delta_{0k} \qquad k = -\frac{(m-1)}{2},\ldots,-1,0,1,\ldots,\frac{(m-1)}{2} \tag{4.3.4}$$

Substituting Eq. (4.3.2) into Eq. (4.3.4), yields:

$$\int_{-\infty}^{\infty} \left[\sum_{i=0}^{n+m} \sum_{j=0}^{n+m} \left(\frac{x}{a}\right)^i \left(\frac{x-x_k}{a}\right)^j \beta_i\beta_j \right] \phi\left(\frac{x}{a}\right)\ \phi\left(\frac{x-x_k}{a}\right) dx = 0 \tag{4.3.5}$$

where $k = -\frac{(m-1)}{2},\ldots,-1,0,1,\ldots,\frac{(m-1)}{2}$. Note that Eq. (4.3.5) implies m equations. With a total of $n+m+1$ equations, an orthogonal window function can easily be obtained by solving $n+m+1$ coefficients β_α.

Next, we will introduce a new set of correction coefficients, γ_α, which enforce the original window function to satisfy the wavelet orthogonality conditions. An orthogonal wavelet function with correction coefficients γ_α is defined as:

$$\psi^*(\frac{x-x_J}{a}) = \left[\gamma_0 + \gamma_1(\frac{x-x_J}{a}) + \gamma_2(\frac{x-x_J}{a})^2 + \ldots + \gamma_m(\frac{x-x_J}{a})^m \right] \phi(\frac{x-x_J}{a}) \tag{4.3.6}$$

where $k = -\frac{(m-1)}{2},\ldots,-1,0,1,\ldots,\frac{(m-1)}{2}$ and 'm' is the number of nodes covered by the support of the window function $\phi(\frac{x}{a})$. Again, these coefficients γ_α can be solved from orthogonality conditions for the wavelet.

Orthogonality Condition for Wavelets

The requirement for an orthogonality between wavelet functions and scaling functions is:

$$\int_{-\infty}^{\infty} \psi^*(x)\phi^*(x-k) = 0 \qquad k = -\frac{(m-1)}{2},\ldots,-1,0,1,\ldots,\frac{(m-1)}{2} \tag{4.3.7}$$

Substituting the definitions of $\psi^*(x)$ and $\phi^*(x-k)$ into Eq. (4.3.7), yields:

$$\int_{-\infty}^{\infty}\left[\sum_{i=0}^{m}\sum_{j=0}^{n+m}\left(\frac{x}{a}\right)^{i}\left(\frac{x-x_k}{a}\right)^{j}\gamma_i\beta_j\right]\phi\left(\frac{x}{a}\right)\,\phi\left(\frac{x-x_k}{a}\right)dx=0 \qquad (4.3.8)$$

Where $k=-\frac{(m-1)}{2},\ldots,-1,0,1,\ldots,\frac{(m-1)}{2}$. Note that Eq. (4.3.8) also implies m equations. Finally, to accomplish the orthogonality of the window function and wavelet function, the determination of the coefficients γ_α and β_α is necessary. Although the coefficients γ_α are entangled with β_α in Eq. (4.3.8), they can be treated as a set of linear equations once β_α is obtained. However, it is still difficult to obtain closed-form solutions for β_α with a set of non-linear equations, Eq. (4.3.5). A numerical solution will be the possible choice for obtaining these coefficients.

4.4 Two Dimensional (2D) RKM

By a direct extension from one dimension, a general 2D window function is defined as:

$$\boldsymbol{\Phi}_{\boldsymbol{a}}(\boldsymbol{x})\equiv\boldsymbol{\Phi}_{(a_{x_1},a_{x_2})}(x_1,x_2) \qquad (4.4.1)$$

A special 2D window function can be represented by the product of 1D window functions. Thus,

$$\boldsymbol{\Phi}_{\boldsymbol{a}}(\boldsymbol{x})\equiv\boldsymbol{\Phi}_{(a_{x_1},a_{x_2})}(x_1,x_2)=\phi_{a_{x_1}}(x_1)\phi_{a_{x_2}}(x_2) \qquad (4.4.2)$$

The basis function for 2D is given in the form of:

$$\boldsymbol{P}^{T}(\boldsymbol{x})=\left[1,\boldsymbol{P}_1^{T}(\boldsymbol{x}),\boldsymbol{P}_2^{T}(\boldsymbol{x}),\ldots,\boldsymbol{P}_k^{T}(\boldsymbol{x})\right] \qquad (4.4.3)$$

where

$$\boldsymbol{P}_k^{T}(\boldsymbol{x})=\left[x_1^{k},x_1^{k-1}x_2,\ldots,x_1x_2^{k-1},x_2^{k}\right] \qquad k\text{th order polynomials} \qquad (4.4.4)$$

Reproducing Conditions in 2D

The reproducing equation for 2D is now defined as:

$$u^{R_a}(\boldsymbol{x}) = \int_{-\infty}^{\infty} u(\boldsymbol{y})\boldsymbol{\Phi}_a(\boldsymbol{x}-\boldsymbol{y})d\boldsymbol{y} \tag{4.4.5}$$

Substituting a 2D Taylor series expansion into Eq. (4.4.5) yields:

$$\begin{aligned} u^{R_a}(\boldsymbol{x}) &= m_{00}(\boldsymbol{a},\boldsymbol{x})u(\boldsymbol{x}) - \left[m_{10}(\boldsymbol{a},\boldsymbol{x})u_{,x_1}(\boldsymbol{x}) + m_{01}(\boldsymbol{a},\boldsymbol{x})u_{,x_2}(\boldsymbol{x})\right] \\ &+ \frac{1}{2}\left[m_{20}(\boldsymbol{a},\boldsymbol{x})\frac{\partial^2}{\partial x_1^2} + 2m_{11}(\boldsymbol{a},\boldsymbol{x})\frac{\partial^2}{\partial x_1 \partial x_2} + m_{02}(\boldsymbol{a},\boldsymbol{x})\frac{\partial^2}{\partial x_2^2}\right]u(\boldsymbol{x}) + \cdots \\ &+ \frac{(-1)^n}{n!}m_{IJ}(\boldsymbol{a},\boldsymbol{x})\left[\frac{\partial}{\partial x_1} + \frac{\partial}{\partial x_2}\right]^n u(\boldsymbol{x}) + \frac{(-1)^{n+1}}{(n+1)!}m_{IJ}(\boldsymbol{a},\boldsymbol{x})\left[\frac{\partial}{\partial x_1} + \frac{\partial}{\partial x_2}\right]^{n+1} u(\bar{\boldsymbol{x}}) \end{aligned} \tag{4.4.6}$$

where a 2D moment equation is defined as:

$$m_{IJ}(\boldsymbol{a}, x_1, x_2) = \int_{-\infty}^{\infty} (x_1 - y_1)^I (x_2 - y_2)^J \boldsymbol{\Phi}_a(\boldsymbol{x}-\boldsymbol{y})d\boldsymbol{y} \qquad 0 \le I,\ J \le k \tag{4.4.7}$$

From Eq. (4.4.6), it can be seen that to reproduce the solution correctly up to *n*th order in 2D, the reproducing conditions require that:

$$m_{IJ}(\boldsymbol{a},\boldsymbol{x}) = \delta_{0I}\delta_{0J} \qquad 0 \le I,\ J \le n \tag{4.4.8}$$

To satisfy the reproducing conditions, the modified window function for 2D is assumed to be:

$$\overline{\boldsymbol{\Phi}}_a(\boldsymbol{x}) = \left[\boldsymbol{P}^T(\boldsymbol{x}-\boldsymbol{y})\boldsymbol{b}(\boldsymbol{a},\boldsymbol{x})\right]\boldsymbol{\Phi}_a(\boldsymbol{x}) \tag{4.4.9}$$

where

$$\boldsymbol{b}(\boldsymbol{a},\boldsymbol{x})^T = \left[b_{00}(\boldsymbol{a},\boldsymbol{x}), b_{10}(\boldsymbol{a},\boldsymbol{x}), b_{01}(\boldsymbol{a},\boldsymbol{x}), \ldots, b_{n0}(\boldsymbol{a},\boldsymbol{x}), \ldots, b_{0n}(\boldsymbol{a},\boldsymbol{x})\right] \tag{4.4.10}$$

Then, the moments of the modified window function in 2D are denoted as:

$$\overline{m}_{IJ}(\boldsymbol{a},\boldsymbol{x}) = \int_{-\infty}^{\infty} (x_1 - y_1)^I (x_2 - y_2)^J \left[\boldsymbol{P}^T(\boldsymbol{x}-\boldsymbol{y})\boldsymbol{b}(\boldsymbol{a},\boldsymbol{x})\right] \boldsymbol{\Phi}_{\boldsymbol{a}}(\boldsymbol{x}-\boldsymbol{y})d\boldsymbol{y}$$

$$= \int_{-\infty}^{\infty} (x_1 - y_1)^I (x_2 - y_2)^J \left[\sum_{r,\,s}^{n} (x_1 - y_1)^r (x_2 - y_2)^s b_{rs}(\boldsymbol{a},\boldsymbol{x})\right] \boldsymbol{\Phi}_{\boldsymbol{a}}(\boldsymbol{x}-\boldsymbol{y})d\boldsymbol{y}$$

$$= \sum_{r,\,s=0}^{n} m_{pq}(\boldsymbol{a},\boldsymbol{x}) b_{rs}(\boldsymbol{a},\boldsymbol{x}) \qquad p = r + I \text{ and } q = s + J \tag{4.4.11}$$

or in matrix notation:

$$\overline{\boldsymbol{m}}(\boldsymbol{a},\boldsymbol{x}) = \boldsymbol{M}(\boldsymbol{a},\boldsymbol{x})\boldsymbol{b}(\boldsymbol{a},\boldsymbol{x}) \tag{4.4.12}$$

where $\boldsymbol{M}(\boldsymbol{a},\boldsymbol{x})$ is the moment matrix in 2D, thus:

$$\boldsymbol{M}(\boldsymbol{a},\boldsymbol{x}) = \begin{bmatrix} m_{00} & m_{10} & m_{01} & \cdots & m_{n0} & \cdots & m_{0n} \\ m_{10} & m_{20} & m_{11} & & \cdots & & m_{1n} \\ m_{01} & m_{11} & m_{02} & & \cdots & & m_{0\;n+1} \\ \vdots & & & & & & \\ m_{n0} & \vdots & \vdots & & \ddots & & \vdots \\ \vdots & & & & & & \\ m_{0n} & m_{1n} & m_{0\;n+1} & & \cdots & & m_{0\;2n} \end{bmatrix} \tag{4.4.13}$$

Enforcing the reproducing conditions Eq. (4.4.8) on Eq. (4.4.11) yields:

$$\boldsymbol{M}(\boldsymbol{a},\boldsymbol{x})\boldsymbol{b}(\boldsymbol{a},\boldsymbol{x}) = \boldsymbol{P}(0) \tag{4.4.14}$$

Then. the coefficient vector $\boldsymbol{b}(\boldsymbol{a},\boldsymbol{x})$ can be computed by matrix inversion

$$\boldsymbol{b}(\boldsymbol{a},\boldsymbol{x}) = \boldsymbol{M}^{-1}(\boldsymbol{a},\boldsymbol{x})\boldsymbol{P}(0) \tag{4.4.15}$$

4.5 Multi-Dimensional RKM

To present RKM in multiple dimensions, multi-index notation is adopted.

Multi-index Notation

We will use multi-index notation in the following. α ($\alpha \equiv [\alpha_1, \alpha_2, \cdots \alpha_N]$) is a multi-index if $\alpha_i (i = 1, 2, \cdots N)$ are non-negative integers with the following properties for this multi-index α:

i $$|\alpha| \equiv \sum_{i=1}^{N} \alpha_i \tag{4.5.1a}$$

ii $$\alpha! \equiv \alpha_1!\,\alpha_2!\cdots\alpha_N! \tag{4.5.1b}$$

We then define two symbols

iii $$\boldsymbol{x}^{\alpha} \equiv x_1^{\alpha_1}\,x_2^{\alpha_2}\cdots x_N^{\alpha_N} \tag{4.5.1c}$$

iv $$D^{\alpha}u(\boldsymbol{x}) \equiv \partial_{x_1}^{\alpha_1}\,\partial_{x_2}^{\alpha_2}\cdots\partial_{x_N}^{\alpha_N}u(\boldsymbol{x}) \tag{4.5.1d}$$

With this multi-index notation, the Taylor series expansion in terms of degree n followed by a remainder can be written as:

$$u(\boldsymbol{y}) = \sum_{|\alpha|=0}^{n} \frac{(-1)^{\alpha}}{\alpha!} D^{\alpha}u(\boldsymbol{x})(\boldsymbol{x}-\boldsymbol{y})^{\alpha} + \sum_{|\alpha|=n+1} \frac{(-1)^{\alpha}}{\alpha!} D^{\alpha}u(\bar{\boldsymbol{x}})(\boldsymbol{x}-\boldsymbol{y})^{\alpha} \tag{4.5.2}$$

To understand how this multi-index notation works for the Taylor series expansion, a 3D example is shown next.

Example 1. 3D Taylor's formula using the multi-index notation.

In 3D, the properties for multi-index notation are (Eq.(4.5.1)):

i $$|\alpha| \equiv \alpha_1 + \alpha_2 + \alpha_3 \tag{4.5.3a}$$

ii $$\alpha! \equiv \alpha_1!\,\alpha_2!\,\alpha_3! \tag{4.5.3b}$$

iii $$(\boldsymbol{x}-\boldsymbol{y})^{\alpha} \equiv (x_1 - y_1)^{\alpha_1}(x_2 - y_2)^{\alpha_2}(x_3 - y_3)^{\alpha_3} \tag{4.5.3c}$$

iv $$D^{\alpha}u(\boldsymbol{x}) \equiv \partial_{x_1}^{\alpha_1}\partial_{x_2}^{\alpha_2}\partial_{x_3}^{\alpha_3}u(\boldsymbol{x}) \tag{4.5.3d}$$

Table 5.1 lists all the parameter values for the multi-index and shows how this multi-index notation works for 3D Taylor's formula.

$\|\alpha\|$	α_1	α_2	α_3	$\alpha!$	$(\boldsymbol{x}-\boldsymbol{y})^{\alpha}$	$D^{\alpha}u(\boldsymbol{x})$	$\sum_{\|\alpha\|=0}^{k}\frac{(-1)^{\alpha}}{\alpha!}D^{\alpha}u(\boldsymbol{x})(\boldsymbol{x}-\boldsymbol{y})^{\alpha}$
0	0	0	0	1	1	$u(\boldsymbol{x})$	$u(\boldsymbol{x})$
1	1	0	0	1	$(x_1 - y_1)$	$\partial_{x_1}u(\boldsymbol{x})$	$-(x_1 - y_1)\partial_{x_1}u(\boldsymbol{x})$
	0	1	0	1	$(x_2 - y_2)$	$\partial_{x_2}u(\boldsymbol{x})$	$-(x_2 - y_2)\partial_{x_2}u(\boldsymbol{x})$
	0	0	1	1	$(x_3 - y_3)$	$\partial_{x_3}u(\boldsymbol{x})$	$-(x_3 - y_3)\partial_{x_3}u(\boldsymbol{x})$
2	2	0	0	2	$(x_1 - y_1)^2$	$\partial_{x_1}^2 u(\boldsymbol{x})$	$\frac{1}{2}\big[(x_1 - y_1)^2\partial_{x_1}^2$
	0	2	0	2	$(x_2 - y_2)^2$	$\partial_{x_2}^2 u(\boldsymbol{x})$	$+(x_2 - y_2)^2\partial_{x_2}^2$
	0	0	2	2	$(x_3 - y_3)^2$	$\partial_{x_3}^2 u(\boldsymbol{x})$	$+(x_3 - y_3)^2\partial_{x_3}^2$
	1	1	0	1	$(x_1 - y_1)(x_2 - y_2)$	$\partial_{x_1}\partial_{x_2}u(\boldsymbol{x})$	$+2(x_1 - y_1)(x_2 - y_2)\partial_{x_1}\partial_{x_2}$
	0	1	1	1	$(x_2 - y_2)(x_3 - y_3)$	$\partial_{x_2}\partial_{x_3}u(\boldsymbol{x})$	$+2(x_2 - y_2)(x_3 - y_3)\partial_{x_2}\partial_{x_3}$
	1	0	1	1	$(x_1 - y_1)(x_3 - y_3)$	$\partial_{x_1}\partial_{x_3}u(\boldsymbol{x})$	$+2(x_1 - y_1)(x_3 - y_3)\partial_{x_1}\partial_{x_3}\big]u(\boldsymbol{x})$
3	3	0	0	6	$(x_1 - y_1)^3$	$\partial_{x_1}^3 u(\boldsymbol{x})$	$-\frac{1}{6}\big[(x_1 - y_1)^3\partial_{x_1}^3$
	0	3	0	6	$(x_2 - y_2)^3$	$\partial_{x_2}^3 u(\boldsymbol{x})$	$+(x_2 - y_2)^3\partial_{x_2}^3$
	0	0	3	6	$(x_3 - y_3)^3$	$\partial_{x_3}^3 u(\boldsymbol{x})$	$+(x_3 - y_3)^3\partial_{x_3}^3$
	2	1	0	2	$(x_1 - y_1)^2(x_2 - y_2)$	$\partial_{x_1}^2\partial_{x_2}u(\boldsymbol{x})$	$+3(x_1 - y_1)^2(x_2 - y_2)\partial_{x_1}^2\partial_{x_2}$
	2	0	1	2	$(x_1 - y_1)^2(x_3 - y_3)$	$\partial_{x_1}^2\partial_{x_3}u(\boldsymbol{x})$	$+3(x_1 - y_1)^2(x_3 - y_3)\partial_{x_1}^2\partial_{x_3}$
	1	2	0	2	$(x_1 - y_1)(x_2 - y_2)^2$	$\partial_{x_1}\partial_{x_2}^2 u(\boldsymbol{x})$	$+3(x_1 - y_1)(x_2 - y_2)^2\partial_{x_1}\partial_{x_2}^2$
	0	2	1	2	$(x_2 - y_2)^2(x_3 - y_3)$	$\partial_{x_2}^2\partial_{x_3}u(\boldsymbol{x})$	$+3(x_2 - y_2)^2(x_3 - y_3)\partial_{x_2}^2\partial_{x_3}$
	0	1	2	2	$(x_2 - y_2)(x_3 - y_3)^2$	$\partial_{x_2}\partial_{x_3}^2 u(\boldsymbol{x})$	$+3(x_2 - y_2)(x_3 - y_3)^2\partial_{x_2}\partial_{x_3}^2$
	1	0	2	2	$(x_1 - y_1)(x_3 - y_3)^2$	$\partial_{x_1}\partial_{x_3}^2 u(\boldsymbol{x})$	$+3(x_1 - y_1)(x_3 - y_3)^2\partial_{x_1}\partial_{x_3}^2$
	1	1	1	1	$(x_1 - y_1)(x_2 - y_2)(x_3 - y_3)$	$\partial_{x_1}\partial_{x_2}\partial_{x_3}u(\boldsymbol{x})$	$+6(x_1 - y_1)(x_2 - y_2)(x_3 - y_3)\partial_{x_1}\partial_{x_2}\partial_{x_3}\big]u(\boldsymbol{x})$

Table 5.1 Terms in 3D Taylors Formula in multi-index notation

Multi-Dimensional RKM

The general window function for *N* dimensions is defined as:

$$\boldsymbol{\Phi}_{\boldsymbol{a}}(\boldsymbol{x}) \equiv \boldsymbol{\Phi}_{(a_{x_1}, a_{x_2}, \cdots, a_{x_N})}(x_1, x_2, \cdots, x_N) \tag{4.5.4}$$

Assume an N-dimensional, n-order, l-components basis function is given as:

$$\boldsymbol{P}^T(\boldsymbol{x}) = \left[1, \boldsymbol{P}_1^T(\boldsymbol{x}), \boldsymbol{P}_2^T(\boldsymbol{x}), \dots, \boldsymbol{P}_n^T(\boldsymbol{x})\right] \tag{4.5.5}$$

where

$$\boldsymbol{P}_n^T(\boldsymbol{x}) = \left[x_1^n, x_1^{n-1}x_2, \dots, x_{N-1}x_N^{n-1}, x_N\right] \tag{4.5.6}$$

Reproducing Conditions in N-Dimensions

The reproducing equation for multiple dimensions are:

$$u^{R_a}(\boldsymbol{x}) = \int_{-\infty}^{\infty} u(\boldsymbol{y})\boldsymbol{\Phi}_{\boldsymbol{a}}(\boldsymbol{x} - \boldsymbol{y})d\boldsymbol{y} \tag{4.5.7}$$

Substitution of Eq. (4.5.2) into Eq. (4.5.7) by using the modified window function $\overline{\boldsymbol{\Phi}}_{\boldsymbol{a}}(\boldsymbol{x};\boldsymbol{x} - \boldsymbol{y})$ gives:

$$u^{R_a}(\boldsymbol{x}) = \sum_{|\alpha|=0}^{n} \partial^\alpha u(\boldsymbol{x})\overline{m}_\alpha(\boldsymbol{a}, \boldsymbol{x}) + \sum_{|\alpha|=n+1} \partial^\alpha u(\overline{\boldsymbol{x}})\overline{m}_\alpha(\boldsymbol{a}, \boldsymbol{x}) \tag{4.5.8}$$

where α is a multi-index, $\partial^\alpha u(\boldsymbol{x}) \equiv \dfrac{(-1)^\alpha}{\alpha!} D^\alpha u(\boldsymbol{x})$ and

$$\overline{m}_\alpha(\boldsymbol{a}, \boldsymbol{x}) = \int_{-\infty}^{\infty} (\boldsymbol{x} - \boldsymbol{y})^\alpha \overline{\boldsymbol{\Phi}}_{\boldsymbol{a}}(\boldsymbol{x} - \boldsymbol{y})d\boldsymbol{y} \tag{4.5.9}$$

The α moment of the modified window function $\overline{\boldsymbol{\Phi}}_{\boldsymbol{a}}(\boldsymbol{x};\boldsymbol{x} - \boldsymbol{y})$ satisfies:

$$\overline{m}_\alpha(\boldsymbol{a}, \boldsymbol{x}) = \delta_{0\alpha} \tag{4.5.10}$$

From previous analysis, the reproducing conditions in Eq. (4.5.10) imply

$$\boldsymbol{M}(\boldsymbol{a},\, \boldsymbol{x})\boldsymbol{b}(\boldsymbol{a},\, \boldsymbol{x}) = \boldsymbol{P}(\boldsymbol{0}) \qquad (4.5.11)$$

where $\boldsymbol{M}(\boldsymbol{a}, \boldsymbol{x})$ consists of moments of the window function of $\overline{\boldsymbol{\Phi}}_{\boldsymbol{a}}(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{y})$, that is:

$$\boldsymbol{M}(\boldsymbol{a},\boldsymbol{x}) = \begin{bmatrix} M_{00} & M_{10} & M_{20} & \cdots & M_{n0} \\ M_{01} & M_{11} & & & M_{n1} \\ M_{02} & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ M_{0n} & M_{1n} & \cdots & \cdots & M_{nn} \end{bmatrix} \qquad (4.5.12)$$

and the components of the moment matrix are:

$$M_{\alpha\beta}(\boldsymbol{a},\boldsymbol{x}) = \int_{-\infty}^{\infty} (\boldsymbol{x}-\boldsymbol{y})^{\alpha}\, \boldsymbol{\Phi}_{\boldsymbol{a}}(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{y})(\boldsymbol{x}-\boldsymbol{y})^{\beta}\, d\boldsymbol{y} \qquad 0 \le |\alpha|, |\beta| \le k \qquad (4.5.13)$$

where both α and β are multi-indices.

Reproducing Conditions of Derivatives in N-Dimensions

Extending the 1D analysis, the reproducing conditions of the γth derivative for N dimensions are given by:

$$\overline{m}_{\alpha.\gamma}(\boldsymbol{a},\, \boldsymbol{x}) = (-1)^{\alpha}\alpha!\delta_{\gamma\alpha} \qquad (4.5.14)$$

These conditions imply:

$$D^{\gamma}\left[\boldsymbol{M}(\boldsymbol{a},\, \boldsymbol{x})\boldsymbol{b}(\boldsymbol{a},\, \boldsymbol{x})\right] = D^{\gamma}\left[\boldsymbol{P}(\boldsymbol{0})\right] \qquad (4.5.15a)$$

or in the matrix form:

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ D\boldsymbol{M} & \boldsymbol{M} & & & \boldsymbol{0} \\ D^{2}\boldsymbol{M} & 2D\boldsymbol{M} & \boldsymbol{M} & & \boldsymbol{0} \\ \vdots & & & \ddots & \vdots \\ C_{\gamma}^{0}D^{\gamma}\boldsymbol{M} & C_{\gamma}^{1}D^{\gamma-1}\boldsymbol{M} & \cdots & \cdots & C_{\gamma}^{\gamma}\boldsymbol{M} \end{bmatrix} \begin{bmatrix} \boldsymbol{b} \\ D\boldsymbol{b} \\ D^{2}\boldsymbol{b} \\ \vdots \\ D^{\gamma}\boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{P}(\boldsymbol{0}) \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \end{bmatrix} \qquad (4.5.15b)$$

where $$C_\gamma^m = \frac{\gamma!}{(\gamma - m)!m!} \qquad (m = 0, 1, 2, \cdots, \gamma) \tag{4.5.16}$$

5. REPRODUCING KERNEL PARTICLE METHODS (RKPM)

The reproducing kernel particle method (RKPM) is the discrete form of RKM. For the discrete system, the reproducing conditions can be obtained following an extension of the procedure developed in the continuous case.

5.1 Reproducing Kernel Particle Method In One Dimension

Reproducing Condition

The reproducing equation in the discrete system is given as:

$$u^{R_a}(x) = \sum_{j=1}^{np} u(x_j)\phi_a(x - x_j)\Delta x_j \tag{5.1.1}$$

where np is the number of the particles in the discrete system. We, then, define the numerical moments of the window function as:

$$\tilde{m}_k(a,x) = \sum_{j=1}^{np} \left(x - x_j\right)^k \phi_a\left(x - x_j\right)\Delta x_j \qquad k = 0, 1, \ldots, n \tag{5.1.2}$$

Recall from Eq. (4.2.3), to reproduce $u^{R_a}(x)$ exactly up to nth order, we require:

$$\tilde{m}_\alpha(a,x) = \delta_{0\alpha} \qquad \alpha = 0, 1, 2, \ldots, n \tag{5.1.3}$$

To satisfy the reproducing conditions Eq. (5.1.3), a modified window function in the discrete system is assumed to be in the form of:

$$\bar{\phi}_a\left(x; x - x_j\right) = \sum_{k=0}^{n} b_k(a,x)\left(x - x_j\right)^k \phi_a\left(x - x_j\right) = \boldsymbol{P}^T\left(x - x_j\right)\boldsymbol{b}(a,x)\phi_a\left(x - x_j\right)$$

$$= \tilde{C}_a\left(x; x - x_j\right)\phi_a\left(x - x_j\right) \tag{5.1.4}$$

The basis function $\boldsymbol{P}^T$ in Eq. (5.1.4) is chosen as a polynomial type vector. The coefficients $b_k(a, x)$ can be solved by applying the modified window function to the reproducing condition. That is:

$$\tilde{\boldsymbol{M}}(a, x)\boldsymbol{b}(a, x) = \boldsymbol{P}(0) \tag{5.1.5}$$

where $\tilde{\boldsymbol{M}}(a, x)$ is the numerical moment matrix:

$$\tilde{\boldsymbol{M}}(a, x) = \begin{bmatrix} \tilde{m}_0(a, x) & \tilde{m}_1(a, x) & \cdots & \tilde{m}_n(a, x) \\ \tilde{m}_1(a, x) & \tilde{m}_2(a, x) & \cdots & \tilde{m}_{n+1}(a, x) \\ \vdots & & & \vdots \\ \tilde{m}_n(a, x) & \tilde{m}_{n+1}(a, x) & \cdots & \tilde{m}_{2n}(a, x) \end{bmatrix}$$

$$= \sum_{j=1}^{np} \boldsymbol{P}\left(x - x_j\right)\phi_a\left(x - x_j\right)\boldsymbol{P}^T\left(x - x_j\right)\Delta x_j \tag{5.1.6}$$

Then, the coefficient vector $\boldsymbol{b}(a, x)$ can be obtained:

$$\boldsymbol{b}(a, x) = \tilde{\boldsymbol{M}}^{-1}(a, x)\boldsymbol{P}(0) \tag{5.1.7}$$

This RKPM can be viewed as a modified convolution formula for a finite domain and nonuniformly sampled system. The modified window function, $\overline{\phi}_a\left(x; x - x_j\right)$, is the low-pass filter for this finite domain, which can be viewed as a nonuniformly sampled system. In computational mechanics, we use this to define the shape function, such that

$$N_j(x) = \tilde{C}_a(x, x_j)\phi_a\left(x - x_j\right)\Delta x \qquad \text{shape function} \tag{5.1.8}$$

and $$u^R(x) = \sum_{j=1}^{np} u(x_j)N_j(x) \qquad \text{reconstruction} \tag{5.1.9}$$

Reproducing Conditions for the Derivative

The γth derivative of the discrete reproducing equation is defined as:

$$D^{\gamma}u^{R_a}(x) = \sum_{j=1}^{np} u(x_j)D^{\gamma}\phi_a\left(x - x_j\right)\Delta x_j \tag{5.1.10}$$

where '*D*' denotes a differential operator, then, the reproducing conditions for the γth derivative can also be obtained:

$$\tilde{m}_{\alpha.\gamma}(a,x) = (-1)^{\alpha}\,\alpha!\,\delta_{\alpha\gamma} \tag{5.1.11}$$

where

$$\tilde{m}_{\alpha.\gamma}(a,x) = \sum_{j=1}^{np} (x - x_j)^{\alpha} D^{\gamma}\phi_a(x - x_j)\Delta x_j \tag{5.1.12}$$

Example 1: From Eq. (5.1.4), to reproduce the first derivative up to the first order accuracy, the modified window function is assumed to be in the form of:

$$\bar{\phi}_a\left(x; x - x_j\right) = \left[b_0(a,x) + b_1(a,x)\left(x - x_j\right)\right]\phi_a\left(x - x_j\right) \tag{5.1.13}$$

Enforcing the reproducing conditions $\tilde{m}_{\alpha}(a,x) = \delta_{0\alpha}$ and the reproducing condition for the derivative $\tilde{m}_{\alpha.1}(a,x) = -\delta_{\alpha 1}$ on the modified window function yields:

$$\begin{bmatrix} m_0' & m_1' & m_0 & m_1 \\ m_1' & m_2' & m_1 & m_2 \end{bmatrix}\begin{bmatrix} b_0 \\ b_1 \\ b_0' \\ b_1' \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{5.1.14}$$

or in the matrix form

$$\tilde{\boldsymbol{M}}'(a,x)\boldsymbol{b}(a,x) + \tilde{\boldsymbol{M}}(a,x)\boldsymbol{b}'(a,x) = \boldsymbol{0} \tag{5.1.15}$$

From Eq. (5.1.15), it can be seen that the reproducing conditions for the first derivative can also be obtained directly by taking the derivative of the reproducing conditions. That is:

$$D^1\left[\tilde{\boldsymbol{M}}(a,x)\boldsymbol{b}(a,x)\right] = D^1\boldsymbol{P}(0) \tag{5.1.16}$$

5.4 Two Dimensional (2D) RKPM

The RKPM can be easily extended to multiple dimensions. A general 2D window function is assumed of the form:

$$\boldsymbol{\Phi}_{\boldsymbol{a}}(\boldsymbol{x}) \equiv \boldsymbol{\Phi}_{(a_x,a_y)}(x,y) \tag{5.2.1}$$

The basis function for 2D is given in the form of:

$$\boldsymbol{P}^T(\boldsymbol{x}) = \left[1, \boldsymbol{P}_1^T(\boldsymbol{x}), \boldsymbol{P}_2^T(\boldsymbol{x}), \ldots, \boldsymbol{P}_k^T(\boldsymbol{x})\right] \tag{5.2.2}$$

where

$$\boldsymbol{P}_k^T(\boldsymbol{x}) = \left[x^k, x^{k-1}y, \ldots, xy^{k-1}, y^k\right] \qquad k\text{th order polynomials} \tag{5.2.3}$$

Reproducing Conditions in 2D

The reproducing equation for 2D is now defined as:

$$u^{R_a}(\boldsymbol{x}) = \sum_{j=1}^{np} u(\boldsymbol{x}_j)\boldsymbol{\Phi}_{\boldsymbol{a}}(\boldsymbol{x}-\boldsymbol{x}_j)\Delta\boldsymbol{x}_j \tag{5.2.4}$$

where $\Delta\boldsymbol{x}_j$ denotes the jth nodal area. The moment in 2D is defined as:

$$\tilde{m}_{IJ}(\boldsymbol{a},x,y) = \sum_{j=1}^{np} (x-x_j)^I(y-y_j)^J\,\boldsymbol{\Phi}_{\boldsymbol{a}}(\boldsymbol{x}-\boldsymbol{x}_j)\Delta\boldsymbol{x}_j \qquad 0 \le I, J \le k \tag{5.2.5}$$

where $k = 0, 1, \ldots, n$. From Eq. (4.4.6), it can be shown that to reproduce the solution correctly up to nth order in 2D, the reproducing conditions are in the form:

$$\tilde{m}_{IJ}(\boldsymbol{a},\boldsymbol{x}) = \delta_{0I}\delta_{0J} \qquad 0 \le I,\ J \le n \tag{5.2.6}$$

The modified window function for 2D is assumed to be:

$$\overline{\boldsymbol{\Phi}}_{\boldsymbol{a}}(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_j) = \left[\boldsymbol{P}^T(\boldsymbol{x}-\boldsymbol{x}_j)\boldsymbol{b}(\boldsymbol{a},\boldsymbol{x})\right]\boldsymbol{\Phi}_{\boldsymbol{a}}(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_j) \tag{5.2.7}$$

where

$$\boldsymbol{b}(\boldsymbol{a},\boldsymbol{x})^T = \left[b_{00}(\boldsymbol{a},\boldsymbol{x}), b_{10}(\boldsymbol{a},\boldsymbol{x}), b_{01}(\boldsymbol{a},\boldsymbol{x}), \ldots, b_{n0}(\boldsymbol{a},\boldsymbol{x}), \ldots, b_{0n}(\boldsymbol{a},\boldsymbol{x})\right] \tag{5.2.8}$$

Enforcing the reproducing conditions Eq. (5.2.6) on the modified window function yields:

$$\tilde{\boldsymbol{M}}(\boldsymbol{a},\boldsymbol{x})\boldsymbol{b}(\boldsymbol{a},\boldsymbol{x}) = \boldsymbol{P}(0) \tag{5.2.9}$$

where $\tilde{\boldsymbol{M}}(\boldsymbol{a},\boldsymbol{x})$, the numerical moment matrix in 2D, is given as:

$$\tilde{\boldsymbol{M}}(\boldsymbol{a},\boldsymbol{x}) = \begin{bmatrix} \tilde{m}_{00} & \tilde{m}_{10} & \tilde{m}_{01} & \cdots & \tilde{m}_{n0} & \cdots & \tilde{m}_{0n} \\ \tilde{m}_{10} & \tilde{m}_{20} & \tilde{m}_{11} & & \cdots & & \tilde{m}_{1n} \\ \tilde{m}_{01} & \tilde{m}_{11} & \tilde{m}_{02} & & \cdots & & \tilde{m}_{0\,n+1} \\ \vdots & & & & & & \\ \tilde{m}_{n0} & \vdots & \vdots & & \ddots & & \vdots \\ \vdots & & & & & & \\ \tilde{m}_{0n} & \tilde{m}_{1n} & \tilde{m}_{0\,n+1} & & \cdots & & \tilde{m}_{0\,2n} \end{bmatrix} \tag{5.2.10}$$

Then, the coefficient vector $\boldsymbol{b}(\boldsymbol{a},\boldsymbol{x})$ can be computed by matrix inversion

$$\boldsymbol{b}(\boldsymbol{a},\boldsymbol{x}) = \tilde{\boldsymbol{M}}^{-1}(\boldsymbol{a},\boldsymbol{x})\boldsymbol{P}(0) \tag{5.2.11}$$

Reproducing Conditions for Derivative in 2D

Extending the 1D analysis, the reproducing conditions of γth derivative for 2D are:

$$\sum_{j=1}^{np} (x-x_j)^{\alpha_1}(y-y_j)^{\alpha_2}\partial^{\gamma_1}\partial^{\gamma_2}\overline{\boldsymbol{\Phi}}_{\boldsymbol{a}}(\boldsymbol{x}-\boldsymbol{x}_j)\Delta\boldsymbol{x}_j = (-1)^{\alpha_1+\alpha_2}\alpha_1!\,\alpha_2!\,\delta_{\alpha_1\gamma_1}\delta_{\alpha_2\gamma_2} \tag{5.2.12}$$

Following the same procedures in Examples 5.1, it can be shown that the reproducing conditions in Eq. (5.2.12) simply imply:

$$D^{\gamma}\tilde{m}(\boldsymbol{a}, \boldsymbol{x}) = D^{\gamma}\left[\tilde{\boldsymbol{M}}(\boldsymbol{a}, \boldsymbol{x})\boldsymbol{b}(\boldsymbol{a}, \boldsymbol{x})\right] \tag{5.2.13}$$

5.5 Three-Dimensional (3D) RKPM

The general window function for *three* dimensions is defined as:

$$\boldsymbol{\Phi}_{\boldsymbol{a}}(\boldsymbol{x}) \equiv \boldsymbol{\Phi}_{(a_x, a_y, a_z)}(x, y, z) \tag{5.3.1}$$

A basis function is given in the form:

$$\boldsymbol{P}^T(\boldsymbol{x}) = \left[1, \boldsymbol{P}_1^T(\boldsymbol{x}), \boldsymbol{P}_2^T(\boldsymbol{x}), \ldots, \boldsymbol{P}_n^T(\boldsymbol{x})\right] \tag{5.3.2}$$

where $\boldsymbol{P}_k^T(\boldsymbol{x})$ are complete polynomials of order k. For example, $\boldsymbol{P}_1^T(\boldsymbol{x}) = [x, y, z]$, $\boldsymbol{P}_2^T(\boldsymbol{x}) = \left[x^2, y^2, z^2, xy, xz, yz\right]$, ..., etc.

Reproducing Conditions for 3D

Recall from 2D case, the reproducing equation for 3D is given as:

$$u^{R_a}(\boldsymbol{x}) = \sum_{j=1}^{np} u(\boldsymbol{x}_j)\boldsymbol{\Phi}_{\boldsymbol{a}}(\boldsymbol{x} - \boldsymbol{x}_j)\Delta\boldsymbol{x}_j \tag{5.3.3}$$

where $\Delta\boldsymbol{x}_j$ denotes the jth nodal volume. The moment in 3D is defined as:

$$\tilde{m}_{IJK}(\boldsymbol{a}, x, y, z) = \sum_{j=1}^{np} (x - x_j)^I (y - y_j)^J (z - z_j)^K \boldsymbol{\Phi}_{\boldsymbol{a}}(\boldsymbol{x} - \boldsymbol{x}_j)\Delta\boldsymbol{x}_j \tag{5.3.4}$$

Recall Eq. (4.5.10), the reproducing conditions are in the form:

$$\tilde{m}_{IJK}(\boldsymbol{a}.\boldsymbol{x}) = \delta_{0I}\delta_{0J}\delta_{0K} \tag{5.3.5}$$

From previous analysis, the reproducing conditions in Eq. (5.3.5) imply

$$\boldsymbol{M}(\boldsymbol{a},\ \boldsymbol{x})\boldsymbol{b}(\boldsymbol{a},\ \boldsymbol{x}) = \boldsymbol{P}(\boldsymbol{0}) \tag{5.3.6}$$

where the moment matrix $\tilde{\boldsymbol{M}}(\boldsymbol{a},\boldsymbol{x})$ for 3D is given by:

$$\tilde{\boldsymbol{M}}(\boldsymbol{a},\boldsymbol{x}) = \begin{bmatrix} \tilde{M}_{00} & \tilde{M}_{10} & \tilde{M}_{20} & \cdots & \tilde{M}_{n0} \\ \tilde{M}_{01} & \tilde{M}_{11} & & & \tilde{M}_{n1} \\ \tilde{M}_{02} & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ \tilde{M}_{0n} & \tilde{M}_{1n} & \cdots & \cdots & \tilde{M}_{nn} \end{bmatrix} \tag{5.3.7}$$

and the components of the numerical moment matrix are:

$$\tilde{M}_{\alpha\beta}(\boldsymbol{a},\boldsymbol{x}) = \sum_{j=1}^{np} (\boldsymbol{x}-\boldsymbol{x}_j)^{\alpha}\, \Phi_{\boldsymbol{a}}(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_j)(\boldsymbol{x}-\boldsymbol{x}_j)^{\beta}\,\Delta\boldsymbol{x}_j \qquad 0 \le |\alpha|, |\beta| \le k \tag{5.3.8}$$

where both α and β are multi-indices.

Reproducing Conditions of Derivative for 3 D

From 2D analysis and with the help of multi-index notation, the reproducing conditions of the γth derivative for 3D are given by:

$$\tilde{m}_{\alpha.\gamma}(\boldsymbol{a},\ \boldsymbol{x}) = (-1)^{\alpha}\,\alpha!\,\delta_{\gamma\alpha} \tag{5.3.9}$$

These conditions imply:

$$D^{\gamma}\left[\tilde{\boldsymbol{M}}(\boldsymbol{a},\ \boldsymbol{x})\boldsymbol{b}(\boldsymbol{a},\ \boldsymbol{x})\right] = D^{\gamma}[\boldsymbol{P}(\boldsymbol{0})] \tag{5.3.10}$$

or in the matrix form:

$$\begin{bmatrix} \tilde{M} & 0 & 0 & \cdots & 0 \\ D\tilde{M} & \tilde{M} & & & 0 \\ D^2\tilde{M} & 2D\tilde{M} & \tilde{M} & & 0 \\ \vdots & & & \ddots & \vdots \\ C_\gamma^0 D^\gamma \tilde{M} & C_\gamma^1 D^{\gamma-1}\tilde{M} & \cdots & \cdots & C_\gamma^\gamma \tilde{M} \end{bmatrix} \begin{bmatrix} b \\ Db \\ D^2 b \\ \vdots \\ D^\gamma b \end{bmatrix} = \begin{bmatrix} P(0) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{5.3.11}$$

where $$C_\gamma^m = \frac{\gamma!}{(\gamma-m)!m!} \qquad (m = 0, 1, 2, \cdots, \gamma) \tag{5.3.12}$$

The 3D computational aspect are given in Appendix A.

6. REPRODUCING KERNEL PARTICLE METHODS IN FOURIER SPACE

6.1 Preliminary

From the Fourier analysis, the convolutions of polynomials of order zero (1), one (x) and two (x^2) with a window function can be expressed as

$$\int_{-\infty}^{+\infty} 1 \cdot \phi_a(x-y)dy = \hat{\phi}(0) \tag{6.1.1}$$

$$\int_{-\infty}^{+\infty} y \cdot \phi_a(x-y)dy = x\hat{\phi}(0) - ia\hat{\phi}'(0) = x\hat{\phi}_a(0) - i\hat{\phi}_a'(0) \tag{6.1.2}$$

$$\int_{-\infty}^{+\infty} y^2 \cdot \phi_a(x-y)dy = x^2\hat{\phi}(0) - i2ax\hat{\phi}'(0) - a^2\hat{\phi}''(0)$$

$$= x^2\hat{\phi}_a(0) - i2x\hat{\phi}_a'(0) - \hat{\phi}_a''(0) \tag{6.1.3}$$

where $\hat{\phi}^{(n)}(0)$ is the nth derivative of $\hat{\phi}(\xi)$ evaluated at x=0

By induction, the convolution of a polynomial of order n with a window function is given as

$$\int_{-\infty}^{+\infty} y^n \cdot \phi(x-y)dy = \sum_{k=0}^{n} (-i)^k C_k^n x^{n-k} a^k \hat{\phi}^{(k)}(0) = \sum_{k=0}^{n} (-i)^k C_k^n x^{n-k} \hat{\phi}_a^{(k)}(0) \tag{6.1.4}$$

and

$$C_k^n = \frac{n!}{(n-k)!k!} \tag{6.1.5}$$

6.2 Reproducing Conditions

In order to reproduce the polynomials correctly, several requirements of the window function need to be established in the Fourier transform domain (ξ):

1. To reproduce [1], we need $\hat{\phi}(0) = 1$. (6.2.6a)
2. To reproduce [1, x], we need $\hat{\phi}(0) = 1$ and $\hat{\phi}'(0) = 0$. (6.2.6b)
3. To reproduce [1, x, x^2], we need $\hat{\phi}(0) = 1$, $\hat{\phi}'(0) = 0$ and $\hat{\phi}''(0) = 0$. (6.2.6c)

Generalizing these reproducing conditions to *n*th order polynomials yields :

To reproduce [1, x, ..., x^n], we need $\hat{\phi}(0) = 1$ and $\hat{\phi}^{(k)}(0) = 0$ for $k \in [1,n]$. (6.2.7)

These reproducing conditions set a constraint on the selection of a window function. The higher the order of polynomial requested, the flatter the window function must be at ξ=0 in the Fourier transform domain. Accordingly, an ideal low-pass-filter window function (a window function with all the derivatives equal to zero at ξ=0) would reproduce a polynomial of arbitrary order.

6.3 Reproducing Conditions in Terms of Window Function Moments

In our research, we are not interested in selecting a window which reproduces a given function, but in designing the window to have certain filtering properties. Therefore, an equivalent expression of the reproducing conditions is needed in the function domain (x). First of all, let us define the moments of the window function as

$$m_k(a,x) = \int_{-\infty}^{+\infty} (x-y)^k \phi_a(x-y)dy \qquad \text{kth moment of window function} \tag{6.3.1}$$

The convolutions defined by moments of a window function are given as

$$\int_{-\infty}^{+\infty} 1 \cdot \phi_a(x-y)dy = m_0(a,x) \tag{6.3.2}$$

$$\int_{-\infty}^{+\infty} y \cdot \phi_a(x-y)dy = \int_{-\infty}^{+\infty} [x-(x-y)] \cdot \phi_a(x-y)dy = xm_0(a,x) - m_1(a,x) \tag{6.3.3}$$

$$\int_{-\infty}^{+\infty} y^2 \cdot \phi_a(x-y)dy = \int_{-\infty}^{+\infty} \left[x^2 - 2x(x-y) + (x-y)^2\right] \cdot \phi_a(x-y)dy$$

$$= x^2 m_0(a,x) - 2xm_1(a,x) + m_2(a,x) \tag{6.3.4}$$

To summarize, the convolution of an *n*th order polynomial is of the form

$$\int_{-\infty}^{+\infty} y^n \cdot \phi_a(x-y)dy = \sum_{k=0}^{n} (-1)^k C_k^n x^{n-k} m_k(a,x) \tag{6.3.5}$$

From Eq.(6.3.5), reproducing conditions similar to Eqs. (6.2.7) can be derived

1. To reproduce [1] we need $m_0(a,x) = 1$. (6.3.6a)
2. To reproduce [1, *x*] we need $m_0(a,x) = 1$ and $m_1(a,x) = 0$. (6.3.6b)
3. To reproduce [1, *x*, x^2] we need $m_0(a,x) = 1$, $m_1(a,x) = 0$ and $m_2(a,x) = 0$ (6.3.6c)

In general, the reproducing conditions for the *n*th order polynomial in the form of moments are given as

To reproduce [1, *x*, ..., x^n] we need $m_0(a,x) = 1$ and $m_k(a,x) = 0$ for $k \in [1,n]$ (6.3.7)

Eq.(6.2.7) and Eq.(6.3.7) can be related by the moment equations

$$m_k(x) = i^k a^k \hat{\phi}^{(k)}(0) = i^k \hat{\phi}_a^{(k)}(0) \quad \text{for } k \in [0,n] \tag{6.3.8}$$

where $i = \sqrt{-1}$.

For most problems, we may obtain a better physical interpretation of the results in the Fourier transform domain (ξ) than in the function domain (x). But to avoid time consuming computations which may involve transforms and inverse transforms, the real computation should be carried out in the function domain (x) only. For the reproducing conditions, Eq.(6.3.8) is the link between the function domain (x) and the Fourier transform domain (ξ). The question of how to design a window function in the Fourier transform domain (ξ) and perform the computation in the function domain (x) is the topic of the next section.

6.4 Design of a Reproducing Kernel Window

Modified Window Function in the Fourier Transform Domain

A given window function may not satisfy the reproducing conditions and we have few clues to find a cure in the function domain (x). However, if we turn our attention to the Fourier transform domain (ξ), we can realize that we can define a modified window function, $\hat{\bar{\phi}}_a(\xi)$ or $\hat{\bar{\phi}}(a\xi)$, as a linear combination of the original window function, $\hat{\phi}_a(\xi)$ or $\hat{\phi}(a\xi)$, and its first and second derivatives, etc. That is;

$$\hat{\bar{\phi}}_a(\xi) = \alpha_0 \hat{\phi}_a(\xi) + \alpha_1 \hat{\phi}_a'(\xi) + \cdots + \alpha_n \hat{\phi}_a^{(n)}(\xi) = \sum_{k=0}^{n} \alpha_k \hat{\phi}_a^{(k)}(\xi) \tag{6.4.1a}$$

or

$$\hat{\bar{\phi}}(a\xi) = \alpha_0 \hat{\phi}(a\xi) + \alpha_1 a \hat{\phi}'(a\xi) + \cdots + \alpha_n a^n \hat{\phi}^{(n)}(a\xi) = \sum_{k=0}^{n} \alpha_k a^k \hat{\phi}^{(k)}(a\xi) \tag{6.4.1b}$$

Now the process to satisfy the reproducing conditions for the modified window function is reduced to determining the proper coefficients α_k, i.e.

$$\begin{bmatrix} \bar{\hat{\phi}}(0) \\ \bar{\hat{\phi}}'(0) \\ \vdots \\ \bar{\hat{\phi}}^{(n)}(0) \end{bmatrix} = \begin{bmatrix} \hat{\phi}(0) & a\hat{\phi}'(0) & \cdots & a^n\hat{\phi}^{(n)}(0) \\ \hat{\phi}'(0) & a\hat{\phi}''(0) & & a^n\hat{\phi}^{(n+1)}(0) \\ \vdots & & & \vdots \\ \hat{\phi}^{(n)}(0) & a\hat{\phi}^{(n+1)}(0) & \cdots & a^n\hat{\phi}^{(2n)}(0) \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{6.4.2}$$

Therefore

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} \hat{\phi}(0) & a\hat{\phi}'(0) & \cdots & a^n\hat{\phi}^{(n)}(0) \\ \hat{\phi}'(0) & a\hat{\phi}''(0) & & a^n\hat{\phi}^{(n+1)}(0) \\ \vdots & & & \vdots \\ \hat{\phi}^{(n)}(0) & a\hat{\phi}^{(n+1)}(0) & \cdots & a^n\hat{\phi}^{(2n)}(0) \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{6.4.3}$$

Modified Window in the Function Domain

The modified window function can be derived in the function domain without applying the inverse Fourier transform. Since the reproducing conditions apply at ξ=0, an equivalent expression can be derived in the function domain (x) by applying the moment equation defined in Eq.(6.3.8), such that

$$\bar{\hat{\phi}}(0) = \sum_{k=0}^{n} \alpha_k a^k \hat{\phi}^{(k)}(0) = \sum_{k=0}^{n} \alpha_k i^{-k} \left[\int_{-\infty}^{+\infty} (x-y)^k \phi_a(x-y)dy \right]$$

$$= \int_{-\infty}^{+\infty} \left[\sum_{k=0}^{n} \alpha_k i^{-k} (x-y)^k \phi_a(x-y)dy \right] = \int_{-\infty}^{+\infty} \bar{\phi}_a(x-y)dy \tag{6.4.4}$$

where $\bar{\phi}(\bullet)$ is the modified window function defined in the function domain (x)

$$\bar{\phi}_a(x-y) = \sum_{k=0}^{n} \alpha_k i^{-k} (x-y)^k \phi_a(x-y) \tag{6.4.5}$$

In general, this modified window function can be expanded in the form of

$$\bar{\phi}_a(x;x-y) = \beta_0(a,x)\phi_a(x-y) + \beta_1(a,x)(x-y)\phi_a(x-y) + \cdots$$

$$+\beta_n(a,x)(x-y)^n \phi_a(x-y)$$

$$=\sum_{k=0}^{n}\beta_k(a,x)(x-y)^k \phi_a(x-y) \qquad (6.4.6)$$

where the coefficients $\beta_k(a,x)$ are associated with the coefficients α_k in the Fourier transform domain (ξ).

List of Modified Window Functions

The proper choice of a window function, which satisfies the reproducing conditions up to the order of polynomial we desire, will make the correction function equal to one. A list of modified window functions, labeled as $\overline{\phi}_a^2(x)$, with the capability to reproduce the [*1, x, x*^{2}] basis without a correction function, i.e. $C_a(x;x\text{-}y)$=1, follow:

Gaussian $$\left[\frac{3}{2}-\left(\frac{x-y}{a}\right)^2\right]\phi_a(x-y) \qquad (6.4.7a)$$

Box function: $$\left[\frac{9}{4}-15\left(\frac{x-y}{a}\right)^2\right]\phi_a(x-y) \qquad (6.4.7b)$$

Hat function: $$\left[\frac{12}{7}-\frac{30}{7}\left(\frac{x-y}{a}\right)^2\right]\phi_a(x-y) \qquad (6.4.7c)$$

Quadratic spline: $$\left[\frac{13}{8}-\frac{5}{2}\left(\frac{x-y}{a}\right)^2\right]\phi_a(x-y) \qquad (6.4.7d)$$

Cubic spline: $$\left[\frac{27}{17}-\frac{30}{17}\left(\frac{x-y}{a}\right)^2\right]\phi_a(x-y) \qquad (6.4.7e)$$

The list of modified functions for the [*1,x, x*^{2},*x*^{3},*x*^{4}] basis, $\overline{\phi}_a^4(x)$, is also given:

Gaussian $$\left[\frac{15}{8}-\frac{5}{2}\left(\frac{x-y}{a}\right)^2+\frac{1}{2}\left(\frac{x-y}{a}\right)^4\right]\phi_a(x-y) \qquad (6.4.8a)$$

Box function: $$\left[\frac{225}{64}-\frac{525}{8}\left(\frac{x-y}{a}\right)^2+\frac{945}{4}\left(\frac{x-y}{a}\right)^4\right]\phi_a(x-y) \qquad (6.4.8b)$$

Hat function: $$\left[\frac{1635}{683}-\frac{10500}{683}\left(\frac{x-y}{a}\right)^2+\frac{11970}{683}\left(\frac{x-y}{a}\right)^4\right]\phi_a(x-y) \tag{6.4.8c}$$

Quadratic spline: $$\left[\frac{144785}{65216}-\frac{66675}{8152}\left(\frac{x-y}{a}\right)^2+\frac{20685}{4076}\left(\frac{x-y}{a}\right)^4\right]\phi_a(x-y) \tag{6.4.8d}$$

Cubic spline: $$\left[\frac{170010}{80347}-\frac{429450}{80347}\left(\frac{x-y}{a}\right)^2+\frac{178290}{80347}\left(\frac{x-y}{a}\right)^4\right]\phi_a(x-y) \tag{6.4.8e}$$

Based on the Fourier analysis, an example comparing a RKPM window function with its corresponding wavelet function and an orthogonal scaling function with its orthogonal wavelet function obtained from section 3 are shown in Figure 6.1. A cubic spline function is used in this example.

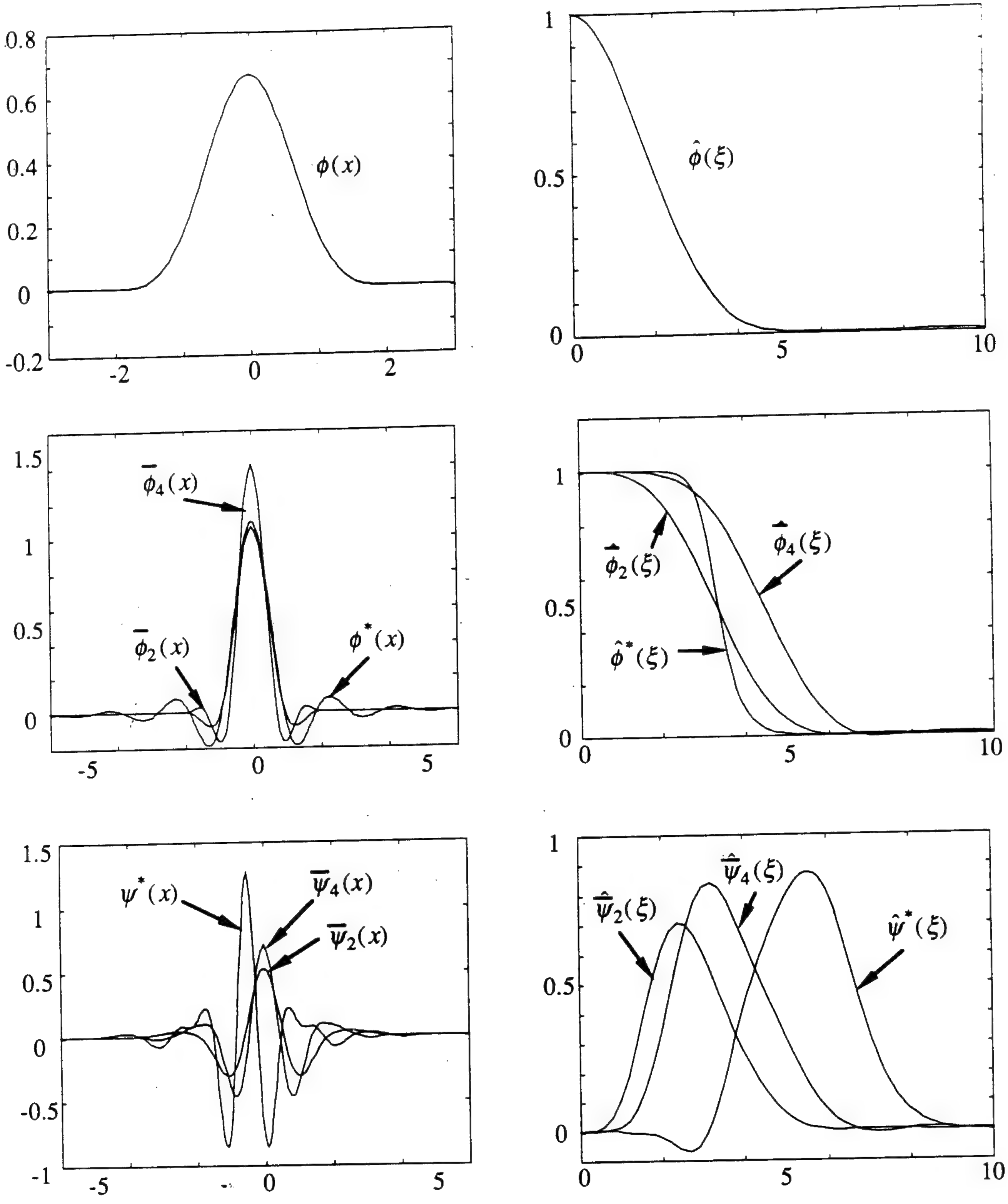


Figure 6.1 Cubic spline function and different modifications in space and Fourier domains

In the frequency domain, the orthogonal scaling function $\hat{\phi}^*(\xi)$ serves as a better low-pass filter than the modified window functions, $\hat{\bar{\phi}}_2(\xi)$ and $\hat{\bar{\phi}}_4(\xi)$. However, the orthogonal scaling function contains more sidelobes in the frequency domain. On the other hand, the orthogonal wavelet function $\hat{\psi}^*(\xi)$ is shown as a band-pass filter in the high frequency region with a sidelobe. With the same frequency range as the orthogonal wavelet function, the modified wavelet functions, $\hat{\bar{\psi}}_2(\xi)$ and $\hat{\bar{\psi}}_4(\xi)$, also serve as band-pass filters but are shifted back from the high frequency range without any sidelobe.

6.5 Boundary Effects from Finite Domain

When applying the convolution formula in a finite domain, a boundary correction technique is needed. Since the convolution formula uses the values of the neighboring points under the support of window function to reproduce the function at a given point, neglecting the values of the function outside the finite domain will cause error. This end-effect can be fixed by enforcing the reproducing conditions in the finite domain. The correction functions change shape when the window function approaches the boundary, compensating for the boundary in the reproducing procedure. Combining the correction function and the window function as the kernel function, RKM is given as a form of a finite domain convolution

$$u^R(x) = \int_\Omega u(y)C_a(x;x,y)\phi_a(x-y)dy = \int_\Omega u(y)\bar{\phi}_a(x;x-y)dy \tag{6.5.1}$$

where $\bar{\phi}_a(x;x-y) = C_a(x;x-y)\phi_a(x-y)$ is the kernel function.

6.6 Consequence of Discretization

Fourier Analysis

The reproducing kernel *particle* method (RKPM) is the discrete form of RKM. The reproducing conditions and the criterion to derive the correction function are different from the forms in the continuous system. As in the continuous case, we will first investigate these conditions in the Fourier transform domain. We start our investigation with uniform sampling. For uniform sampling, choose

$$\Delta x_j = \Delta x \text{ and } x_j = n\Delta x \qquad j \in Z = [\cdots,-1,0,1,\cdots] \tag{6.6.1}$$

The discrete convolution and its equivalent expression is given as

$$\sum_{j=-\infty}^{+\infty} 1\cdot\phi_a\left(x-x_j\right)\Delta x = \sum_{m\in Z}\hat{\phi}_a\left(\frac{2m\pi}{\Delta x}\right)e^{i\left(\frac{2m\pi x}{\Delta x}\right)} = \hat{\phi}_a(0)+(APET)_0 \qquad (6.6.2)$$

where APET stands for the amplitude and phase error terms and it is in the form of

$$(APET)_0 = \sum_{m\in Z^{\pm}}\hat{\phi}_a\left(\frac{2m\pi}{\Delta x}\right)e^{i\left(\frac{2m\pi x}{\Delta x}\right)} \qquad Z^{\pm}=[\cdots,-2,-1,1,2,\cdots]=-I\cup I \qquad (6.6.3)$$

Note that, the leading term of Eq. (6.6.2) is identical to that of the continuous form. The $(APET)_0$ is the additional error introduced by the discretization procedure.

The reproducing condition for linear function gives

$$\sum_{j=-\infty}^{+\infty} x_j\cdot\phi_a\left(x-x_j\right)\Delta x = x\hat{\phi}_a(0)-i\hat{\phi}'_a(0)+x(APET)_0-i(APET)_1 \qquad (6.6.5)$$

where

$$(APET)_1 = \sum_{m\in Z^{\pm}}\hat{\phi}'_a\left(\frac{2m\pi}{\Delta x}\right)e^{i\left(\frac{2m\pi x}{\Delta x}\right)} \qquad (6.6.4)$$

Similarly, the discrete convolution of a quadratic function (x^2) is given as

$$\sum_{j=-\infty}^{+\infty} x_j^2\cdot\phi_a\left(x-x_j\right)\Delta x = x^2\hat{\phi}_a(0)-ix\hat{\phi}'_a(0)+a^2\hat{\phi}''_a(0)$$

$$+x^2(APET)_0-ix(APET)_1+(APET)_2 \qquad (6.6.6)$$

where

$$(APET)_2 = \sum_{m\in Z^{\pm}} \hat{\phi}_a''\left(\frac{2m\pi}{\Delta x}\right) e^{i\left(\frac{2m\pi x}{\Delta x}\right)} \tag{6.6.7}$$

By induction, the reproducing formula for an nth order polynomial yields

$$\sum_{j=-\infty}^{+\infty} x_j^n \cdot \phi_a\left(x-x_j\right)\Delta x = \sum_{k=0}^{n} (-i)^k C_k^n x^{n-k} \hat{\phi}_a^{(k)}(0) + \sum_{k=0}^{n} (-i)^k C_k^n x^{n-k} (APET)_k \tag{6.6.8}$$

where

$$(APET)_k = \sum_{m\in Z^{\pm}} \hat{\phi}_a^{(k)}\left(\frac{2m\pi}{\Delta x}\right) e^{i\left(\frac{2m\pi x}{\Delta x}\right)} \tag{6.6.9}$$

Comparing the continuous and the discrete forms, we find that the differences are in the APETs. From the discrete Fourier analysis, we know that the APETs are the outcome of the system discretization. For the general case, the APETs decrease as the dilation parameter increases, but the APETs can not be eliminated from the reproducing process. Therefore, we recommend a quickly decaying window function to reduce the APETs and to hopefully reduce the error.

7. INTERPOLANT ESTIMATION AND CONVERGENCE OF REPRODUCING KERNEL PARTICLE GALERKIN METHODS

7.1 Error Estimation of Reproducing Kernel Methods

The Taylor's expansion of $u(y)$ with respect to a given position x in the RKM formula gives

$$\begin{aligned} u^{R_a}(x) &= \int_{-\infty}^{+\infty} u(y)\overline{\phi}_a(x-y)dy \\ &= u(x)\int_{-\infty}^{+\infty} \overline{\phi}_a(x-y)dy - u'(x)\int_{-\infty}^{+\infty}(x-y)\overline{\phi}_a(x-y)dy + \cdots \\ &\quad + \frac{(-1)^n}{n!}u^{(n)}(x)\int_{-\infty}^{+\infty}(x-y)^n\overline{\phi}_a(x-y)dy \\ &\quad + \frac{(-1)^{n+1}}{(n+1)!}u^{(n+1)}(x)\int_{-\infty}^{+\infty}(x-y)^{n+1}\overline{\phi}_a(x-y)dy + h.o.t. \end{aligned} \tag{7.1.1}$$

where *h.o.t.* denotes the higher order terms of the Taylor's expansion. The equivalent expression of Eq. (7.1.1) by the definition of window moments is given as

$$\begin{aligned} u^{R_a}(x) &= u(x)\overline{m}_0(x) - u'(x)\overline{m}_1(x) + \cdots + \frac{(-1)^n}{n!}u^{(n)}(x)\overline{m}_n(x) \\ &+ \frac{(-1)^{n+1}}{(n+1)!}u^{(n+1)}(x)\overline{m}_{n+1}(x) + h.o.t. \end{aligned} \tag{7.1.2}$$

From the Fourier analysis of RKM and the reproducing requirement of RKM, the moments of the window function can also be defined as

$$\overline{m}_k(x) = \int_{-\infty}^{+\infty} (x-y)^k \overline{\phi}_a(x-y)dy = i^k \hat{\overline{\phi}}_a^{(k)}(0) \tag{7.1.3}$$

and

$$\overline{m}_k(x) = \int_{-\infty}^{+\infty} (x-y)^k \overline{\phi}_a(x-y)dy = \delta_{0k} \qquad \text{for } k = 1,\cdots,n \tag{7.1.4}$$

The equivalent expression of Eq.(7.1.2) is in the form of

$$u^{R_a}(x) = u(x) + \frac{(-1)^{n+1}}{(n+1)!}u^{(n+1)}(x)\overline{m}_{n+1}(x) + h.o.t. \tag{7.1.5}$$

Therefore, the error of RKM can be defined as

$$\begin{aligned} error(x) &= u^{R_a}(x) - u(x) \\ &= \frac{(-1)^{n+1}}{(n+1)!}u^{(n+1)}(x)\overline{m}_{n+1}(x) + h.o.t. \\ &= \frac{(-1)^{n+1}}{(n+1)!}u^{(n+1)}(x)\left[i^{n+1}\hat{\overline{\phi}}_a^{(n+1)}(0)\right] + h.o.t. \\ &= \frac{(-1)^{n+1}}{(n+1)!}u^{(n+1)}(x)\left[i^{n+1}a^{n+1}\hat{\overline{\phi}}^{(n+1)}(0)\right] + h.o.t. \end{aligned} \tag{7.1.6}$$

Correspondingly, the interpolation estimate can be given by the following theorem.

Theorem 7.1.1 Assume $u(x), \phi(x) \in C^{n+1}(\Omega) \cap H^{n+1}(\Omega)$, where Ω is the domain of interest, then

$$\left\| u - u^{R_a} \right\|_{H^k} \le C_1 a^{n+1-k} \left\| \phi^{'} \right\|_{H^k} |u|_{W_\infty^{n-1}} \left\| \bar{C}(a) \right\|_{H^k} \tag{7.1.7}$$

where a is the dilation parameter, $\phi(x)$ is the window function, $\bar{C}(a)$ is the kth derivative of the associated correction function, C_1 is a constant which is independent of the dilation parameter. For RKM, $\left\|\bar{C}(a)\right\|_{H^k}$ is independent of a; whereas for RKPM, $\left\|\bar{C}(a)\right\|_{H^k}$ approaches a constant when the dilation parameter a is large enough. Note that ϕ' is defined as $\phi' = E(a)\phi$, in which $E(a)$ is a normalization factor and $\int_\Omega \phi' d\Omega = 1$.

In particular for $k = 1$ and 0, we have

$$\left\| u - u^{R_a} \right\|_{H^1} \le C_1 a^{n} \left\| \phi^{'} \right\|_{H^1} |u|_{W_\infty^{n+1}} \left\| \bar{C}(a) \right\|_{H^1} \tag{7.1.8}$$

and

$$\left\| u - u^{R_a} \right\|_{L^2} \le C_2 a^{n+1} \left\| \phi^{'} \right\|_{L^2} |u|_{W_\infty^{n+1}} \left\| \bar{C}(a) \right\|_{L^2} \tag{7.1.9}$$

Proof: From Eq.(4.2.17-19), the derivative error obtained from the Taylor expansion can be rewritten as

$$D^k u^{R_a}(x) - D^k u(x) = -\sum_{\alpha=n+1} \partial^\alpha u(\bar{x}) \int_\Omega (x-y)^\alpha D^k \left\{ \phi_a(x;x-y) C(x;x-y) \right\} dy \tag{7.1.10}$$

Taking the H^k norm on both sides and bounding each term with the maximum value yields

$$\left\| u - u^{R_a} \right\|_{H^k} \le C_1 a^{n+1-k} \left\| \phi^{'} \right\|_{H^k} |u|_{W_\infty^{n+1}} \left\| \bar{C}(a) \right\|_{H^k} \tag{7.1.11}$$

For instance, $u^{(n+1)}(\bar{x})$ can be bounded by $|u|_{W_\infty^{n+1}}$ from definition. The details for this bounding procedure can be found in Liu et al. (1995e).

7.2 Convergence of Reproducing Kernel Galerkin Method

Applying the above interpolation estimate Eq. (7.1.7), we examine the convergence property of the reproducing kernel Galerkin method. To clarify the concepts, we study a simple example of membrane on a elastic foundation . The strong form of the problem is

$$-\nabla^2 u + u = f \qquad x \in \Omega \tag{7.2.1}$$

The weak or variational form of Eq. (7.2.1) is: find $u \in \mathcal{S}$, such that for all $v \in \mathcal{V}$

$$\mathrm{a}(v,u) = (v,f) \tag{7.2.2}$$

where u is the trial function, v is the test function, f is the force term, $\mathcal{S}$ and $\mathcal{V}$ is the collection of trial functions and test functions, respectively. The assembly operators $\mathrm{a}(v,u)$ and (v,f) are defined as $\int_\Omega (\nabla v \cdot \nabla u + vu) d\Omega$ and $\int_\Omega v f d\Omega$, respectively. Similarly, the reproducing kernel Galerkin approximation can be stated as: find $u^a \in \mathcal{S}^a$, such that for all $v^a \in \mathcal{V}^a$

$$\mathrm{a}(v^a,u^a) = (v^a,f) \tag{7.2.3}$$

u^a, v^a are Galerkin approximations which belong to $\mathcal{S}^a$ and $\mathcal{V}^a$, here $\mathcal{S}^a$ and $\mathcal{V}^a$ are subsets of $\mathcal{S}$ and $\mathcal{V}$. The superscript a denotes the dilation parameter which dictates the convergence of the reproducing kernel Galerkin approximation.

Remark: There are distinct differences between u^a and u^{R_a}. Recall from Eq. (5.1.9), $u^{R_a} = \sum_{J=1} N_J u_J$, which is the reconstruction of function u. It can be viewed as the interpolation of sampling values of $u(x_J)$ denoted by u_J; while u^a is the reproducing kernel Galerkin approximation, which can be defined by $u^a = \sum_{J=1} N_J d_J$ and d_J are the nodal coefficients derived from Eq.(7.2.3).

Theorem 7.2.1 Assume $\mathrm{u}, \phi \in C^{n+1} \cap H^{n+1}$, the H^1 convergence rate of $\left\| u^a - u \right\|$ is governed by

$$\left\| u^a - u \right\|_{H^1} \leq C_3 a^n \left\| \phi' \right\|_{H^1} |u|_{W_{-}^{m+1}} \left\| \tilde{C}(a) \right\|_{H^1} \tag{7.2.4}$$

where C_3 is a constant which is independent of dilation parameter a.

Proof: Let $e = u^a - u$ denote the error in the reproducing kernel particle Galerkin approximation. since $\mathcal{V}^a \subset \mathcal{V}$, we may take $v = v^a$ in the variational form Eq. (7.2.2) which gives

$$\mathrm{a}(v^a, u) = (v^a, f) \tag{7.2.5}$$

Subtracting this equation from Eq. (7.2.3) and using the bilinearity of $\mathrm{a}(\cdot,\cdot)$, yield the well-known orthogonality of the error,

$$\mathrm{a}(v^a, e) = 0 \tag{7.2.6}$$

To acquire the desired convergence equation, we may expand $\mathrm{a}(v^a + e, v^a + e)$ as follows

$$\underbrace{\mathrm{a}(v^a + e, v^a + e)}_{\geq 0} = \underbrace{\mathrm{a}(v^a, v^a)}_{\geq 0} + \underbrace{2\mathrm{a}(v^a, e)}_{=0} + \underbrace{\mathrm{a}(e, e)}_{\geq 0} \tag{7.2.7}$$

By virtue of Eq.(7.2.6) and the positive definiteness property of $\mathrm{a}(\cdot,\cdot)$, Eq. (7.2.7) implies

$$\mathrm{a}(e, e) \leq \mathrm{a}(v^a + e, v^a + e) \tag{7.2.8}$$

Now let $v^a = u^{R_a} - u^a$ according to the definition of the variation space $\mathcal{V}$. Thus $v^a + e = u^{R_a} - u^a + u^a - u = u^{R_a} - u$, which immediately results in

$$\mathrm{a}(e, e) \leq \mathrm{a}(u^{R_a} - u, u^{R_a} - u) \tag{7.2.9}$$

By definition, $\mathrm{a}(e, e) = \left\| u^a - u \right\|_{H^1}^2$ and $\mathrm{a}(u - u^{R_a}, u - u^{R_a}) = \left\| u - u^{R_a} \right\|_{H^1}^2$, then

$$\left\| u^a - u \right\|_{H^1}^2 \leq \left\| u - u^{R_a} \right\|_{H^1}^2 \tag{7.2.10}$$

Thus we complete the proof of theorem 7.2.1,

$$\left\| u^a - u \right\|_{H^1} \leq C_3 a^n \left\| \phi' \right\|_{H^1} \left| u \right|_{W_\infty^{m+1}} \left\| \tilde{C}(a) \right\|_{H^1} \tag{7.2.11}$$

by combing Eq.(7.2.10) with theorem 7.1.1.

Remarks:

1. The energy norm of the error is now bounded and the error estimation and convergence can be evaluated based on this theorem.
2. The convergence study for problems involving essential boundary conditions needs further investigation.

7.3 Relationship between Interpolant Estimation and Wavelet Solution

From Eq. (7.1.5), the projected solutions of RKM in the a- scale and the $2a$-scale are given as

$$u^{R_a}(x)=u(x)+\frac{(-1)^{n+1}}{(n+1)!}u^{(n+1)}(x)\left[i^{n+1}a^{n+1}\hat{\bar{\phi}}^{(n+1)}(0)\right]+h.o.t.^{a} \tag{7.3.1}$$

$$u^{R_{2a}}(x)=u(x)+\frac{(-1)^{n+1}}{(n+1)!}u^{(n+1)}(x)\left[i^{n+1}(2a)^{n+1}\hat{\bar{\phi}}^{(n+1)}(0)\right]+h.o.t.^{2a} \tag{7.3.2}$$

The RKM wavelet solution in $2a$-scale is in the form of

$$w^{R_{2a}}(x)=\left(2^{n+1}-1\right)\frac{(-1)^{n+1}}{(n+1)!}u^{(n+1)}(x)\left[i^{n+1}a^{n+1}\hat{\bar{\phi}}^{(n+1)}(0)\right]+\left(h.o.t.^{2a}-h.o.t.^{a}\right) \tag{7.3.3}$$

Neglect the h.o.t.'s, the wavelet solution is proportional to the error term given in Eq. (7.1.6)

$$w^{R_{2a}}(x)\approx\left(2^{n+1}-1\right)error(x) \tag{7.3.4}$$

Eq. (7.3.4) reveals a possibility that the wavelet solution can be an index for the error estimation of the computed solution. We apply this error estimate in adaptivity which is described next.

8 MULTIRESOLUTION FOR REPRODUCING KERNEL METHODS AND h-ADAPTIVITY

8.1 Multiple Scale RKM

The RKM is a modified convolution formulation through a modified window function, such as

$$u^{R_a}(x) = \int_{-\infty}^{+\infty} u(y)\overline{\phi}_a(x-y)dy \tag{8.1.1}$$

where

$$\overline{\phi}_a(x-y) = \left[\beta_0(x) + \beta_1(x)(x-y) + \cdots + \beta_n(x)(x-y)^n\right]\phi_a(x-y) \tag{8.1.2}$$

and n is the order of polynomial which the RKM can reproduce exactly. The Fourier transformation of the RKM formula yields

$$\begin{aligned}\hat{u}^{R_a}(\xi) &= \hat{u}(\xi)\hat{\overline{\phi}}_a(\xi)\\ &= \hat{u}(\xi)\left\{\alpha_0\hat{\phi}_a(\xi) + i\alpha_1\hat{\phi}_a'(\xi) + \cdots + i^n\alpha_n\hat{\phi}_a^{(n)}(\xi)\right\}\\ &= \alpha_0\hat{u}(\xi)\hat{\phi}_a(\xi) + i\alpha_1\hat{u}(\xi)\hat{\phi}_a'(\xi) + \cdots + i^n\alpha_n\hat{u}(\xi)\hat{\phi}_a^{(n)}(\xi)\end{aligned} \tag{8.1.3}$$

where the derivatives of window function, $\hat{\phi}_a'^{(n)}(\xi)$, can be treated as different filters which represent different scales of resolution in the RKM formulation. If we consider the resolution limit of every individual filter, an equivalent expression of the filtered response can be rewritten as

$$\hat{u}(\xi)\hat{\phi}_a^{(k)}(\xi) = \hat{u}_k(\xi)\hat{\phi}_a^{(k)}(\xi) \tag{8.1.4}$$

where $\hat{u}_k(\xi)$ is the scale of the response which matches the resolution limit of the given filter $\hat{\phi}_a^{(k)}(\xi)$. Applying the relation of Eq. (8.1.4) to the RKM formulation, the given response can be separated as

$$\hat{u}^{R_a}(\xi) = \alpha_0\hat{u}_0(\xi)\hat{\phi}_a(\xi) + i\alpha_1\hat{u}_1(\xi)\hat{\phi}_a'(\xi) + \cdots + i^n\alpha_n\hat{u}_n(\xi)\hat{\phi}_a'^{(n)}(\xi) \tag{8.1.5}$$

The inverse Fourier transformation of Eq. (8.1.5) is given in the form of

$$u^{R_a}(x) = \beta_0(x)\int_{-\infty}^{+\infty} u_0(y)\phi_a(x-y)dy + \beta_1(x)\int_{-\infty}^{+\infty} u_1(y)(x-y)\phi_a(x-y)dy$$

$$+\cdots+\beta_n(x)\int_{-\infty}^{+\infty} u_n(y)(x-y)^n \phi_a(x-y)dy \tag{8.1.6}$$

where $u_k(x)$ is the inverse Fourier transformation of $\hat{u}_k(\xi)$ and represents the k-scale of the given response $u(x)$.

From the Fourier transform analysis, we note that this kernel function can be treated as a low-pass filter in the reconstruction procedure. The multiple scale RKPM is defined by a family of kernel functions. The wavelets corresponding to these kernel functions are defined by:

$$\psi_{m+1}(x) = \phi_m(x) - \phi_{m+1}(x) \tag{8.1.7}$$

Therefore, a multiple scale decomposition of a given response is given as:

$$\begin{aligned}
u^h(x) &= u_0(x) && \text{finest scale}\\
&= w_1(x) + u_1(x) && \text{two-level decomposition}\\
&= w_1(x) + w_2(x) + u_2(x) && \text{three-level decomposition}\\
&= \sum_{i=1}^{m} w_i(x) + u_m(x) && \text{m-level decomposition}
\end{aligned} \tag{8.1.8}$$

where

$$u_m(x) = \int_V C(2^m a_0, x, y)\phi_m(x-y)u(y)dy \tag{8.1.9}$$

$$w_m(x) = \int_V \psi_m(x-y)u(y)dy \tag{8.1.10}$$

with

$$\psi_m(x-y) = C(2^{m-1}a_0, x, y)\phi_{m-1}(x-y) - C(2^m a_0, x, y)\phi_m(x-y) \tag{8.1.11}$$

The multiple scale RKPM starts from the finest scale. The subsequent levels of the response are obtained by the decomposition algorithm in Eq.(8.1.8). Because of the multiple scale characteristics of the time-frequency and space-wave number basis for representing functions, the physical interpretation of the computed results is apparent.

8.2 Multiresolution Analysis and Edge Detection

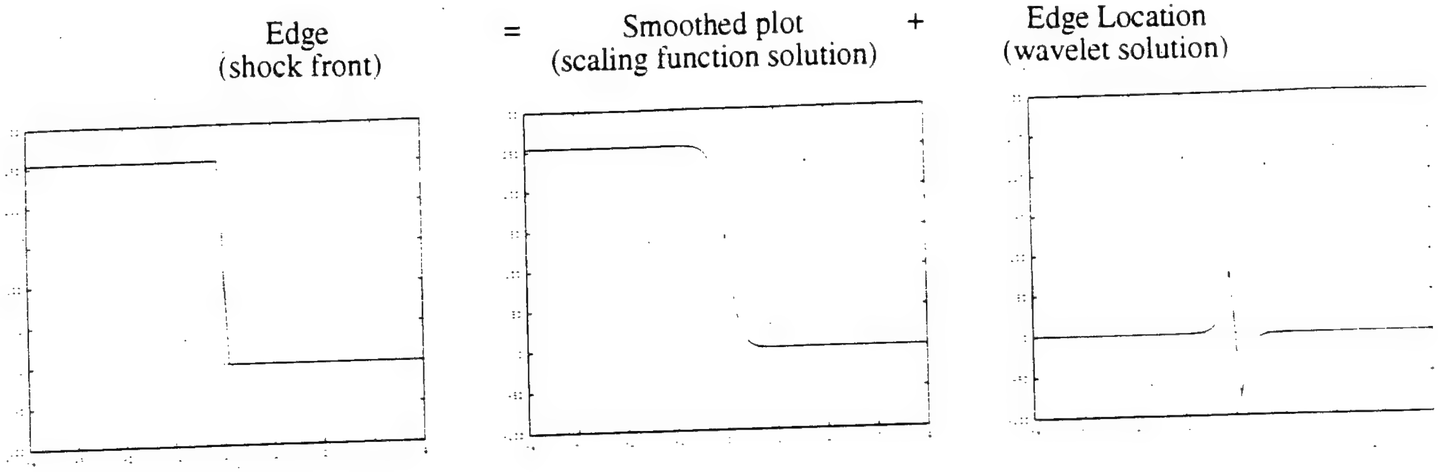


Figure 8.1 Multiresolution Analysis of Shock Front Structure

Digital edge detection is an important technique in data compression and image special effects. For most images, edges define the distinctive features of the picture. With the proper technique, an image can be recovered from its edges without losing significant qualities. Therefore, in data compression, the edges of images are used to optimize the storage space of the system and to reduce the translation cost for network data exchange. Edges can also create special effects in images. A blurred image can be sharpened by adding distinct edges. On the other hand, a picture can be smeared by deleting corresponding edges. While the applications of edge detection keep growing, the mathematical definition of edges remains rather primitive, with most researchers still favoring ad hoc theories In recent years, wavelet theory has become the theory of choice in edge detection, and research in this area has been quite extensive.

In this section, we combine multiresolution analysis from RKPM wavelet theory with the zero-crossing technique to define the edges in images. The procedure is illustrated by a simple 1-D edge structure depicted in Fig.8.1. Based on the multiresolution analysis, the edge structure is decomposed into two different scales: the low scale (the scaling function solution) and the high scale (the wavelet solution). The low scale (scaling function) solution contains the smooth part of the structure, while the high scale (wavelet) solution provides the information for edge detection. In our research, the edges of the image are defined as the zero-crossing locations in the wavelet solution. In this 1-D case, the zero-crossing position is located between the crest and trough of the wavelet like structure in the high scale (wavelet) solution. The multiresolution analysis of a 2-D circle image is shown in Fig.8.2. The low scale (scaling function) solution provides an out-offocus or blurred image of the original while the high scale wavelet solution depicts the outline (edge) of the circle (image).

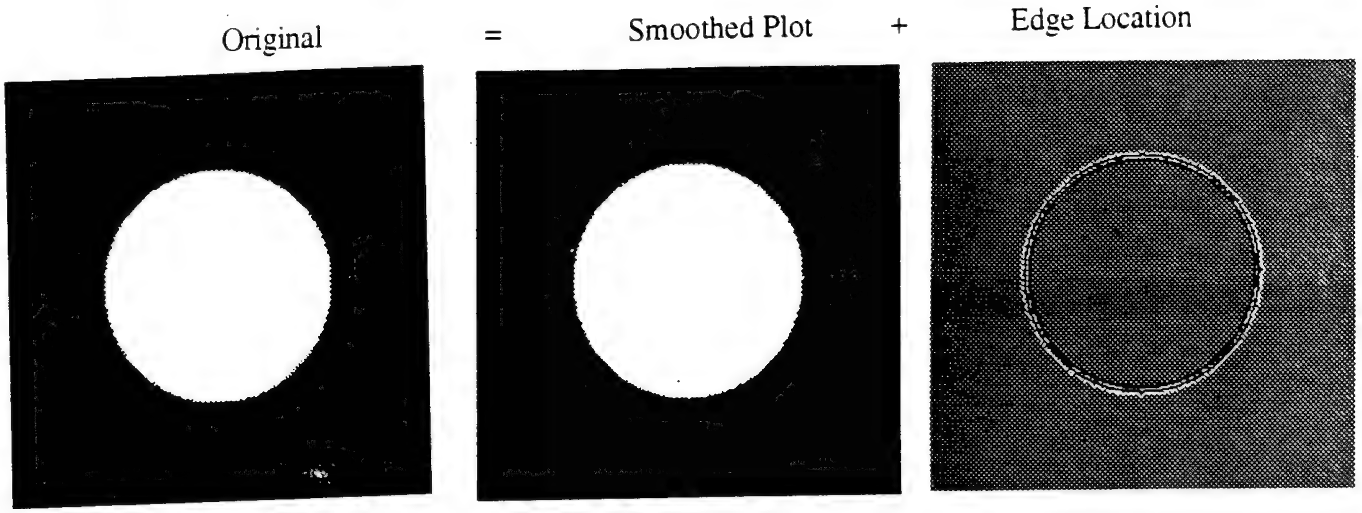
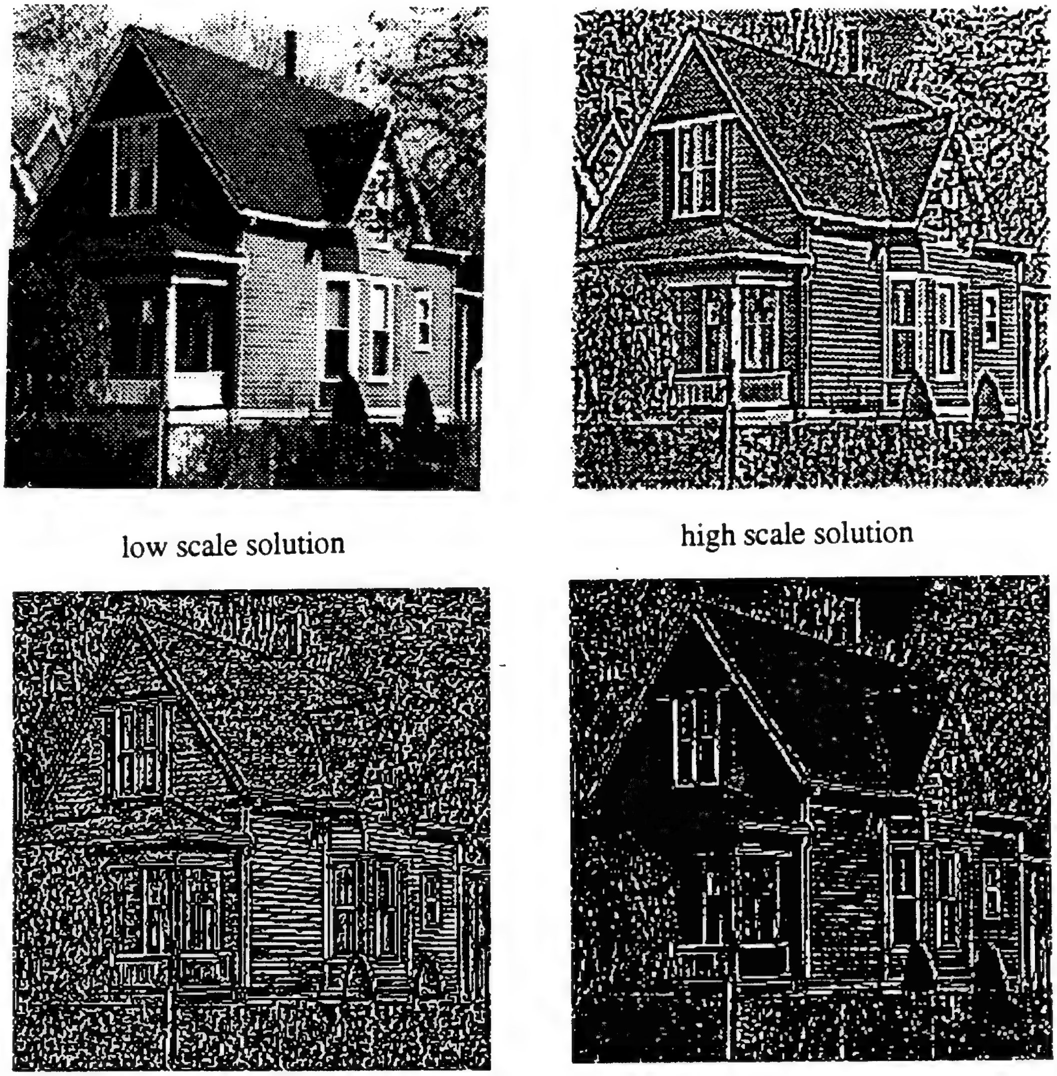


Figure 8.2 Multiresolution Analysis of RKPM in Edge Detection

A two dimensional edge detection example is shown in Fig.8.3. In the first decomposition, the image of a house is separated into low and high (wavelet) scales. Applying the information of the wavelet scale and giving 0% and 10% thresholds, different outcomes are observed.



low scale solution

high scale solution

without threshold

with 10% threshold

Figure 8.3 Edge detection in image processing

9. APPLICATIONS OF RKPM

9.1 Large Deformation Problems

Ring contact

A nonlinear elastic ring contact problem was studied to illustrate the capability of RKPM for large deformation analysis. The material of the rubber ring is Mooney-Rivlin rubber with density ρ=1.4089E-4 ($slug/in^3$). Other material properties are $C_1 = 2000.59 psi$, $C_2 = 200.367 psi$ and $\lambda = 4E6$ *psi*. The rubber ring impacts the wall with initial velocity of 1000 (in/sec). The problem statement and the numerical results obtained at selected time instants are shown in Fig. 9.1.

Bar Impact

The impact of a two-dimensional nonlinear elastic bar was analyzed by RKPM. The material for the bar is Mooney-Rivlin rubber with density ρ=1.4089E-4 (slug). Other material properties are $C_1 = 180.59 psi$, $C_2 = 14.67 psi$ and $\lambda = 1.47E5$ *psi*. A particle distribution of 33 by 9 and timestep of 5E-7 sec are used for computation. Figure 9.2 shows the numerical results of selected time instants when the bar moves toward the wall with initial velocity of 3000 in/sec.

9.2 Computational Fluid Dynamics (CFD)

Under Water Bubble Explosion

Using only a set of particles and Lagrangian RKPM, an underwater explosion problem was solved. Numerical results describing the evolution of the expanding bubble are shown in Figure 9.3. In the computational implementation, the cutoff pressure for water was set to zero and the initial internal pressure 8.72E9 (Pa) is used. The evolution of the expanding bubble is also indicated for selected times. For larger initial internal pressure case, the expansion of the bubble evolved more rapidly so that smaller timestep should be used to capture the whole process.

Thin Biconvex Airfoil Approximation

Two-dimensional, inviscid, compressible flows over a thin biconvex airfoil are examined. Three different Mach numbers 0.5, 0.84 and 1.4 are used to simulate subsonic, transonic and supersonic flows, respectively. Due to the symmetry, only the upper half of the airfoil is considered for computation. Uniform free stream is imposed as inflow boundary conditions and no values are prescribed for the outflow boundary conditions. For a certain region on the surface of the airfoil, $|x| \leq 0.5$, $v = -4bxu_{\infty}$ is specified considering perturbation, where b is the relative thickness. Press contours for each Mach number are shown in Figure 9.4. Aliasing control was conducted for comparison and it can be seen that solutions are improved greatly for transonic and supersonic flows. Also, 2*2 multiple scale decomposition for subsonic and supersonic flows and 3*3 multiple scale decomposition for transonic flow was performed to separate the wavelet solution, which can locate the high gradient region more clearly.

Multiresolution Analysis and Adaptivity

An application of the multiresolution analysis for edge detection and adaptivity refinement in computational fluid dynamics is illustrated in Figure 9.5. In this multiresolution analysis, the numerical solution is examined through error estimation and convergence studies. In the multilevel refinements, only a set of nodes are required for adaptivity with no apprehension of compatibility problems. In addition, the physical interpretation of the numerical solutions can be synthesized by multiple scale representation.

9.3 Structural Acoustics

Plane Wave Scattering Problems

An incident wave $\phi^{(i)}$ travels from left to right along the x-axis (θ=0) with a rigid circle cylinder appearing on its path. The geometry and problem setup are shown in Fig.9.6a.

The solution domain is the shaded area between the inner boundary, representing the cylinder, and the outer artificial boundary, simulating the infinite domain. The first order DtN boundary condition,

$$\left(-i\kappa + \frac{\partial}{\partial r} + \frac{1}{2r}\right)\phi = 0$$

is used in the computation. And the rigid surface reflection has the form:

$$\phi_{,n} + \phi_{,n}^{(i)} = 0 \qquad \text{on } \Gamma_h$$

The series solution is given as:

$$\phi = -2\sum_{n=0}^{\infty} i\frac{J_n'(\kappa a)}{H_n^{(1)'}(\kappa a)}H_n^{(1)}(\kappa r)\cos(n\theta)$$

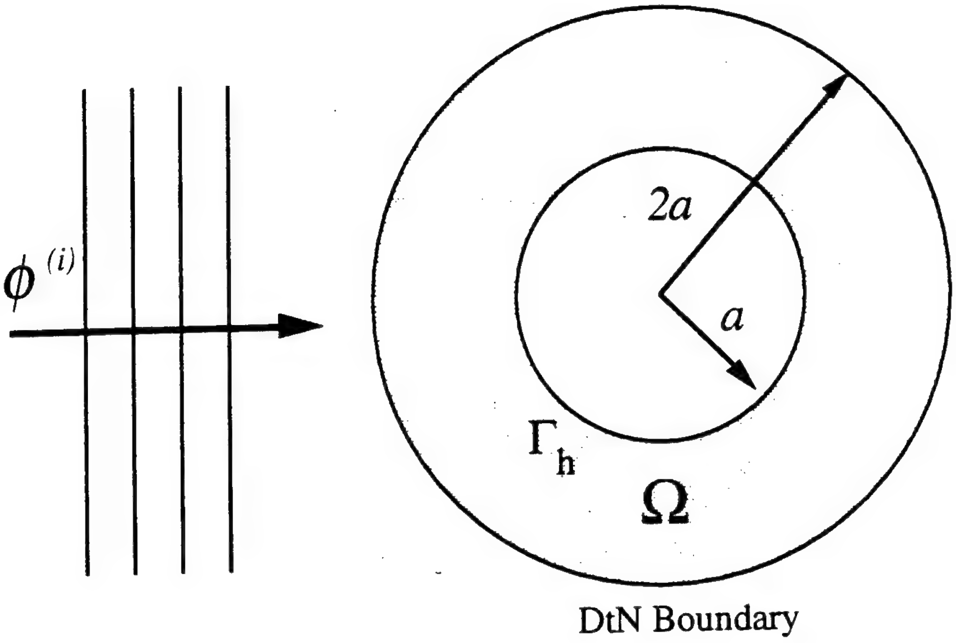


Figure 9.6a Rigid infinite circular cylinder

The numerical results are shown in Fig.9.6b. Note that the edge detection technique is also used to conduct hp-adaptivity in this example.

Multiple Scale Analysis

The flowchart of multiple scale analysis for structural acoustics is outlined in Fig.9.7. By introducing an adequate filter, original solutions can be separated into low scale and high scale solutions (also known as wavelet solutions), which will provide better physical interpretation of the results. Applying the zero-crossing technique described in Section 8.2 on the wavelet solution, the edges of the response can be determined. Since the edges represent the high gradient region of the solution, they automatically provide the locations for local refinement. To enhance the resolution in high gradient region, extra particles are inserted on the edges. The corresponding window functions are adjusted to match the new distribution of the particles. This procedure is similar to the classical hp-adaptive refinement and the behavior of the solutions is also similar. This kind of refinement can continue without encountering any difficulties until the resolution limit is reached, i.e. desired solution is obtained. A numerical example was studied using this procedure and the results are given in Fig.9.7. For the starting level, 684 nodal points were used. Four levels of refinements were performed to acquire the satisfactory results using 2407 nodal points.

Green's Function for a Rectangular Domain

Consider the problem of finding the Green's function with homogeneous Dirichlet boundary condition within a rectangular domain (see figure 9.8a).

$$\nabla^2 u + \kappa^2 u = f$$

where $\kappa^2 = (\omega/c)^2$ and $f = \delta(x - x_0)\delta(y - y_0)$ ($x_0 = y_0 = 0.5$)with boundary conditions

$$u(0, y) = u(L, y) = u(x, 0) = u(x, L) = 0$$

The series solution to this problem is given as:

$$u(x, y) = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} \frac{\sin\left(\frac{m\pi x_c}{L}\right)\sin\left(\frac{m\pi y_c}{L}\right)}{\left(\frac{m\pi}{L}\right)^2 + \left(\frac{n\pi}{L}\right)^2 - \kappa^2} \sin\left(\frac{m\pi x}{L}\right)\sin\left(\frac{n\pi y}{L}\right)$$

Note that there exist natural frequencies whenever

$$\kappa^2 = \left(\frac{m\pi}{L}\right)^2 + \left(\frac{n\pi}{L}\right)^2$$

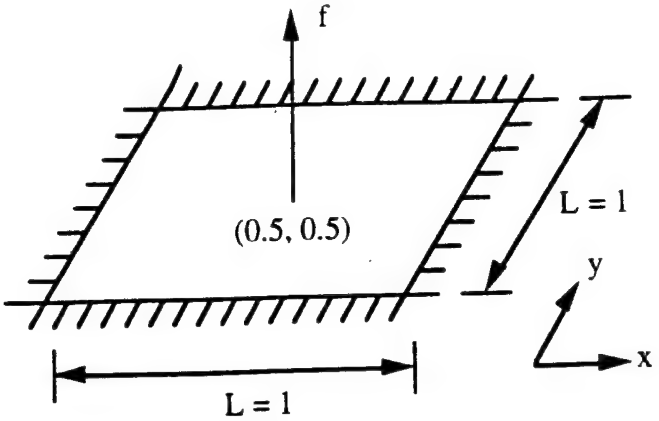


Figure 9.8a Rectangular Domain

A uniform 25x25 particles with mesh size Δx=Δy=1/24 is used for RKPM solution. It is equivalent a 24x24 linear FEM mesh, a 12x12 quadratic FEM mesh, or a 8x8 cubic FEM mesh. The cubic spline function is used as the window function in the RKPM with dE/E=0.1% as the aliasing control parameter [Liu and Chen (1995)]. The resolution limit is κ (wave number)=75.40 or $\pi/(1/24)$. The L_2-norm of the series and numerical solutions with the wave numbers swept from 0 to 80 are shown in Fig. 9.8b. When the wave number is approached to the natural frequency of the system, we can observe a peak in the L_2-norm plot. If the discrete system is a good approximation of the continuous system, the peaks from two systems will match point to point in the frequency (wave number) domain. In the low wave number region, those peaks match quiet well for the most of the cases, but begin to fail to match with the continuous system in higher wave number region. The position where the peaks begin to separate will determine the resolvable scale of the given system (the combination effect of discretization and interpolation). As depicted in the plots, the linear finite element fails quite early, and then follows the quadratic FE. The cubic FE does not survive long enough either. The RKPM can extend its resolvable wave number up to 60, almost 80% (60/75.4) of the resolution limit. In other words, we can use only 3.5 points to catch a cos-sin wave exactly by RKPM (the resolution limit is 3.14 points!).

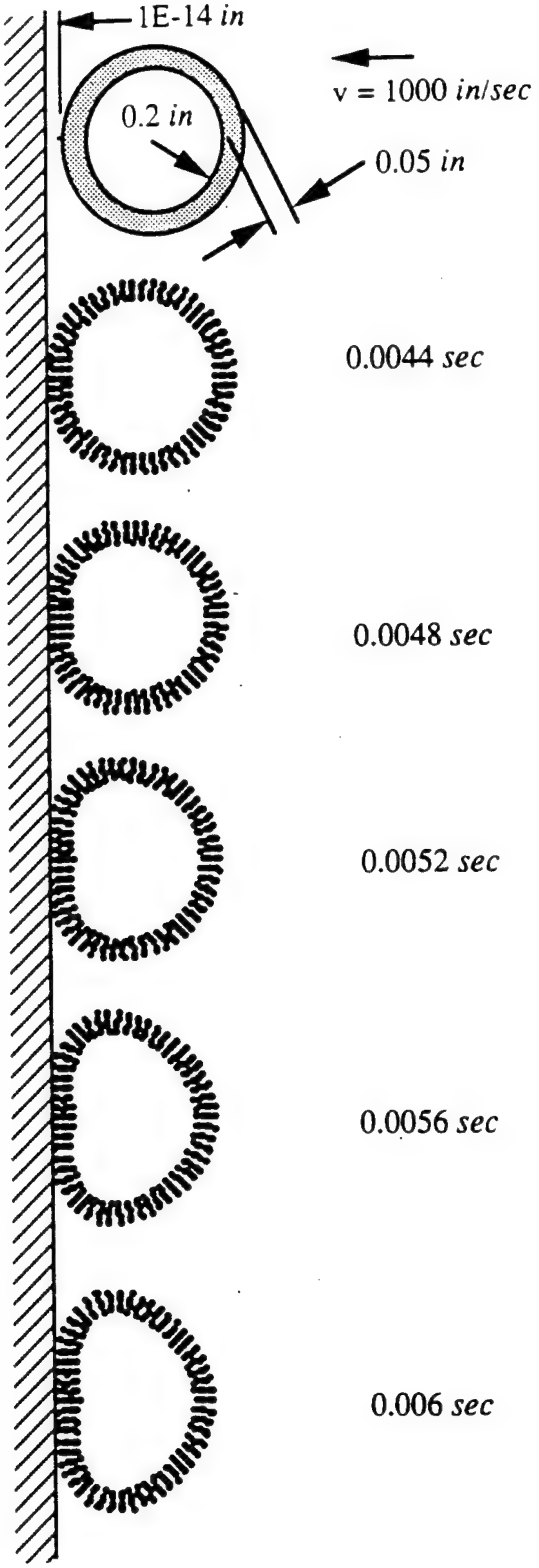


Figure 9.1 Ring contact for large deformation problem

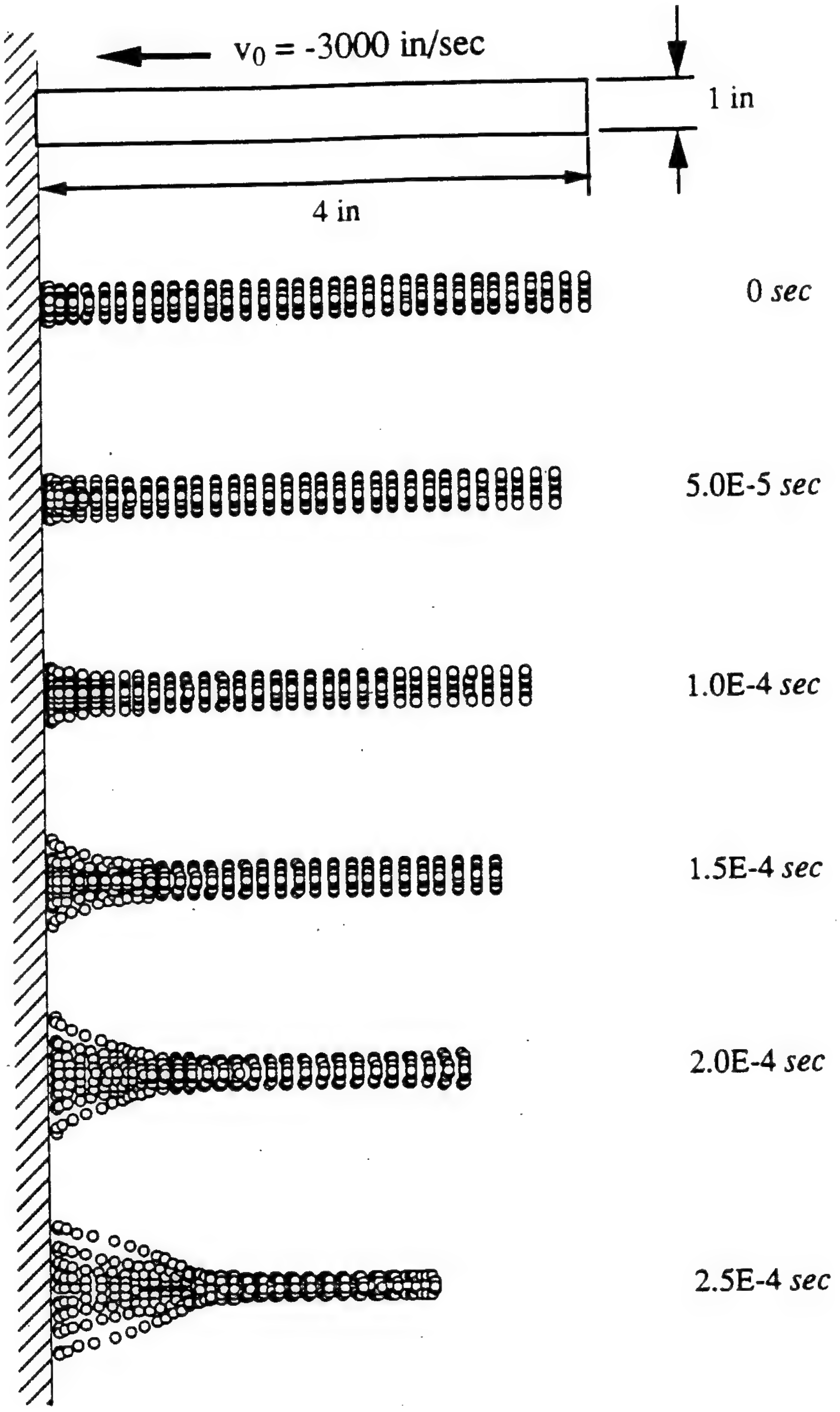


Figure 9.2 Bar impact for large deformation problem

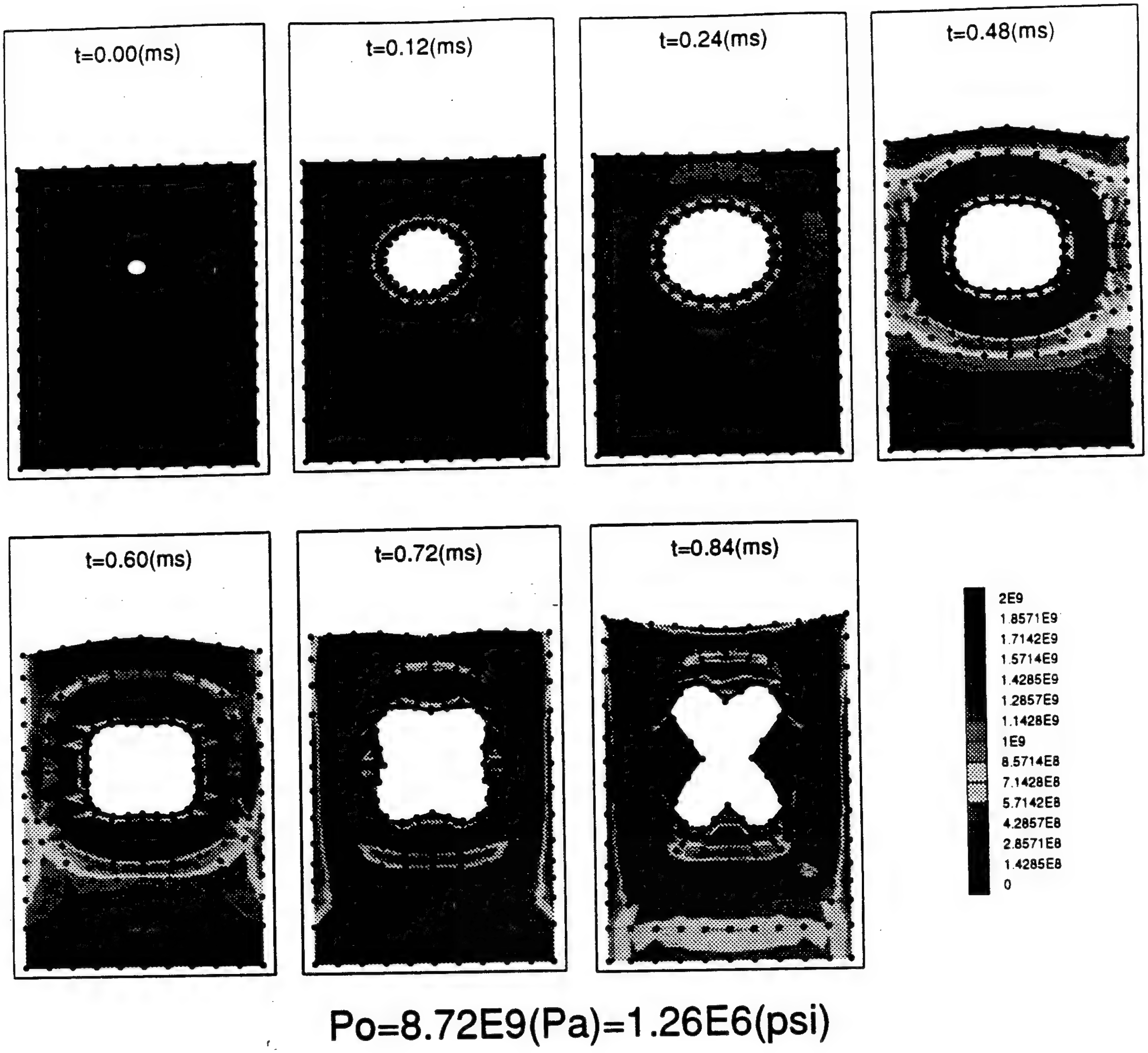
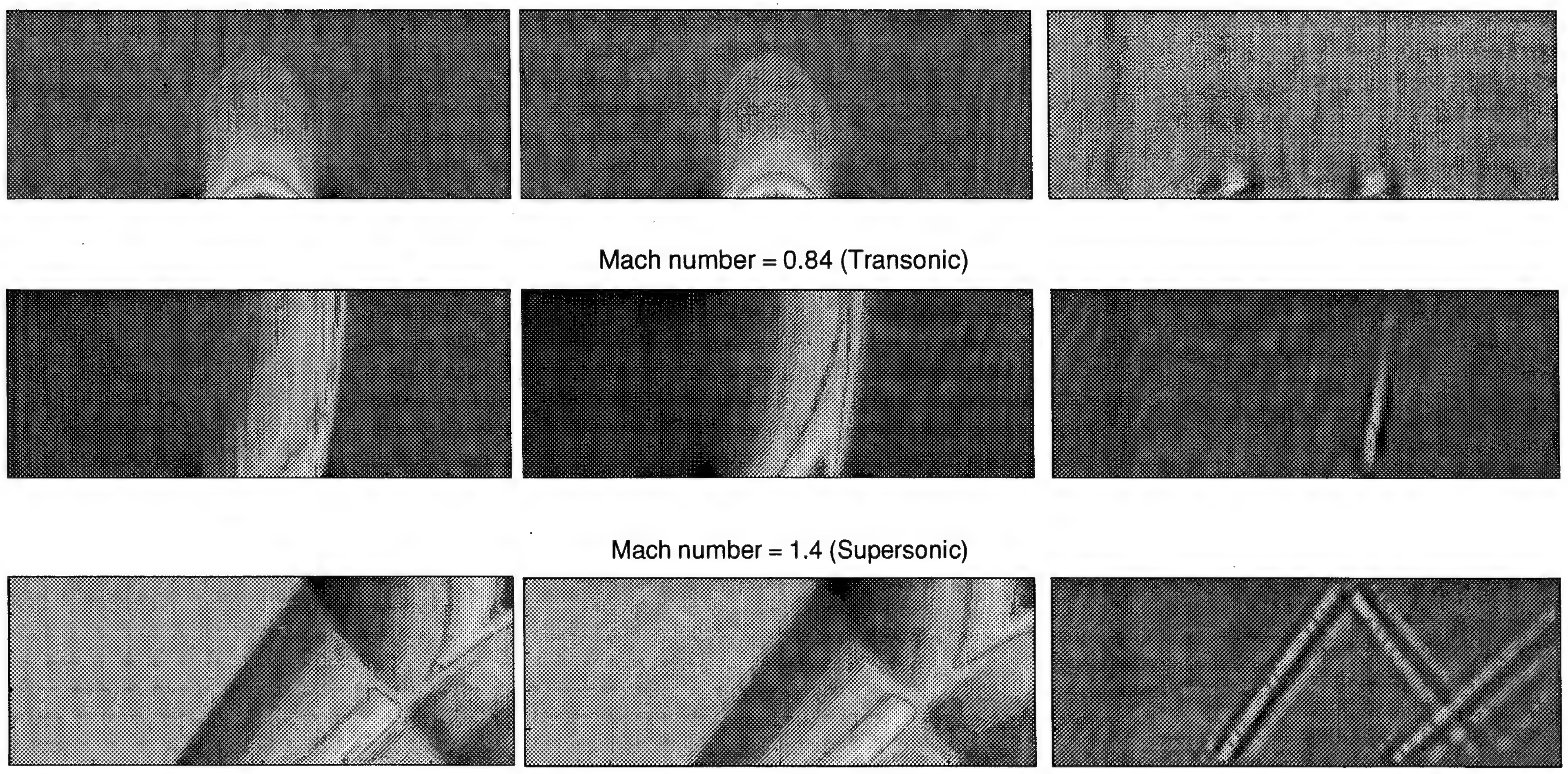


Figure 9.3 Under water bubble explosion



Mach number = 0.5 (Subsonic)

Mach number = 0.84 (Transonic)

Mach number = 1.4 (Supersonic)

RKPM Solutions Without Aliasing Control

RKPM Solutions With Aliasing Control

high wave number band (wavelet) solutions depicting the shock (or high gradient) locations

Figure 9.4 Thin biconvex airfoil approximation

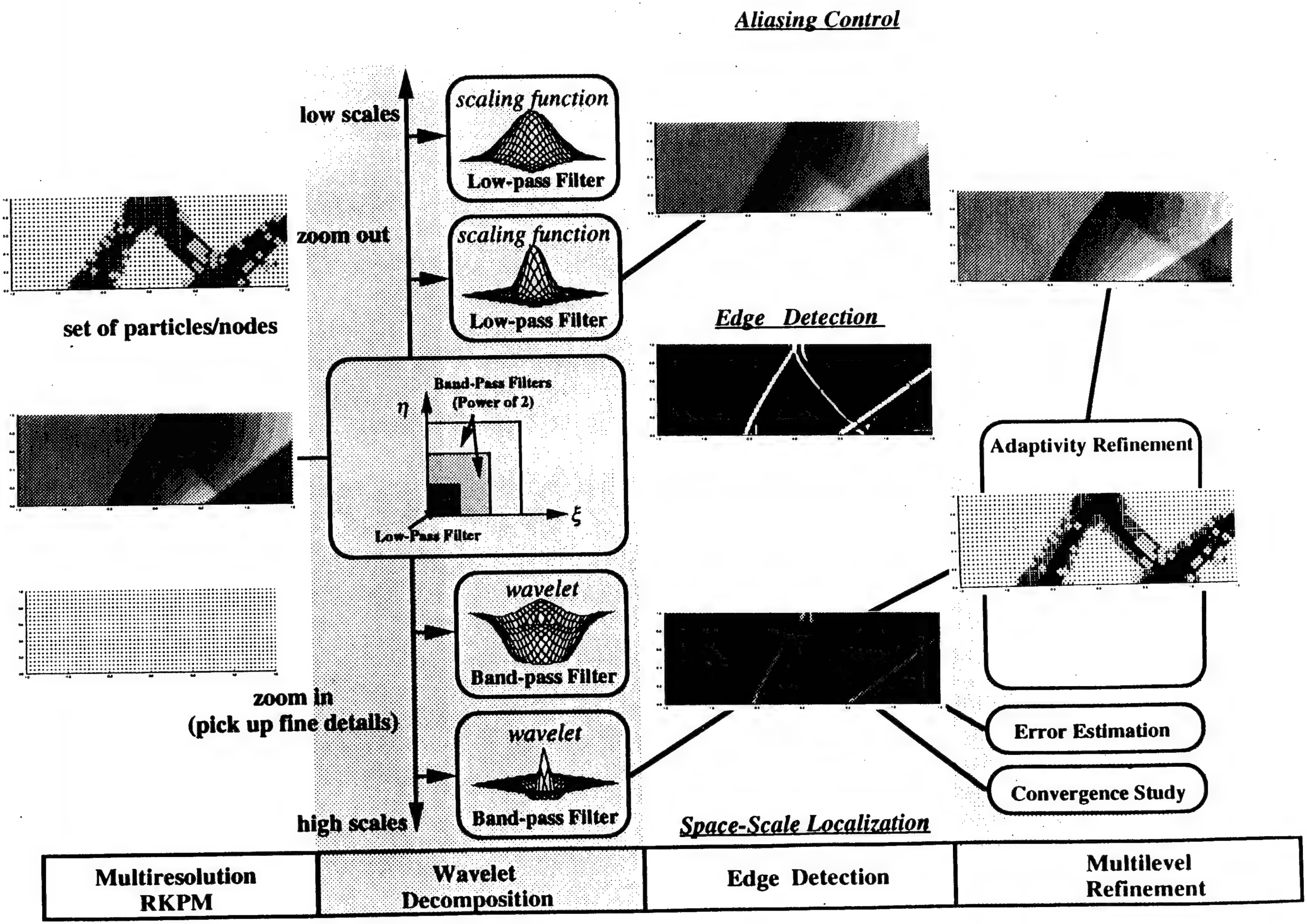
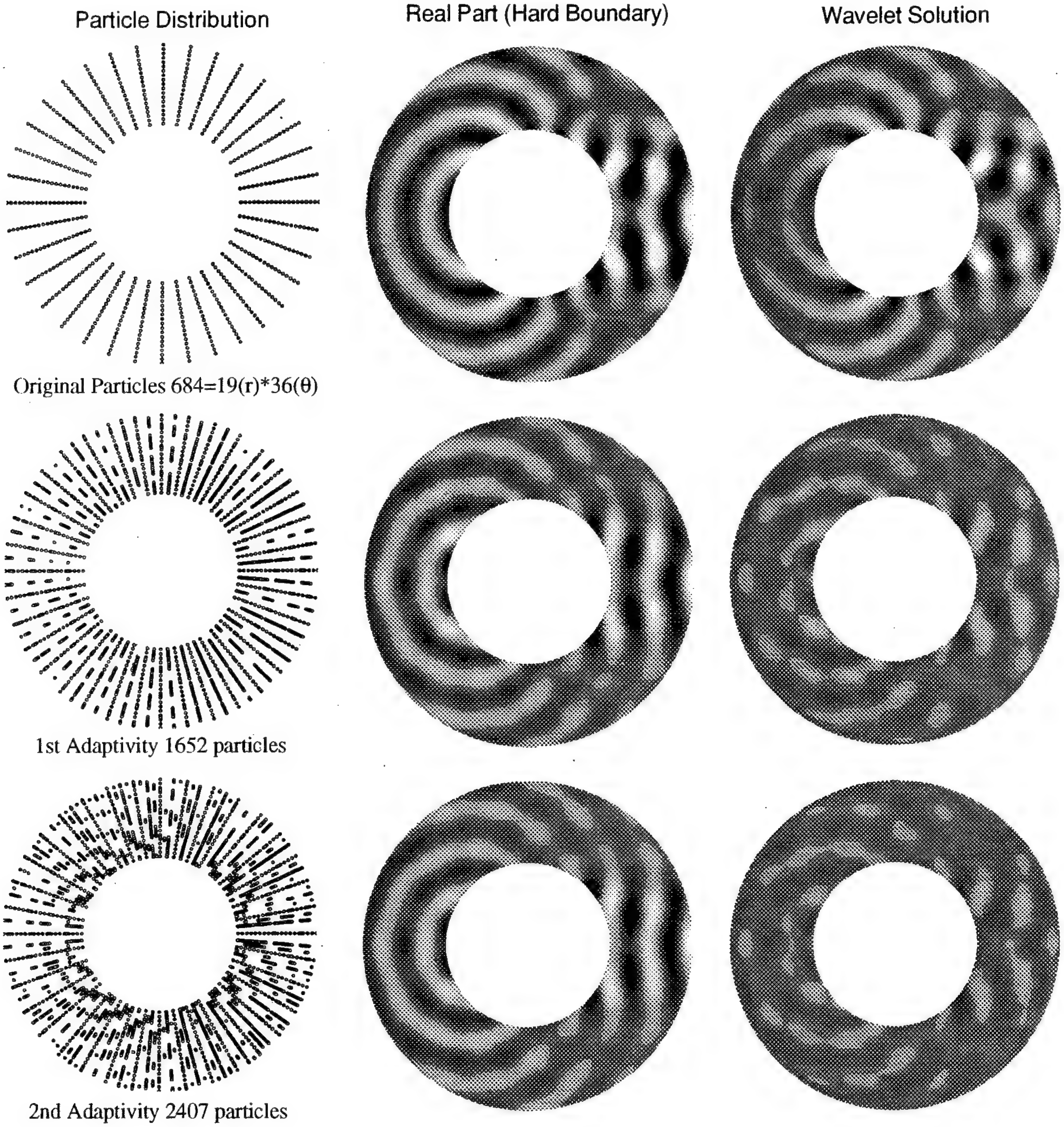


Figure 9.5 Multiresolution analysis and adaptivity for CFD



Particle Distribution

Real Part (Hard Boundary)

Wavelet Solution

Original Particles 684=19(r)*36(θ)

1st Adaptivity 1652 particles

2nd Adaptivity 2407 particles

Figure 9.6b Plane wave scattering problems

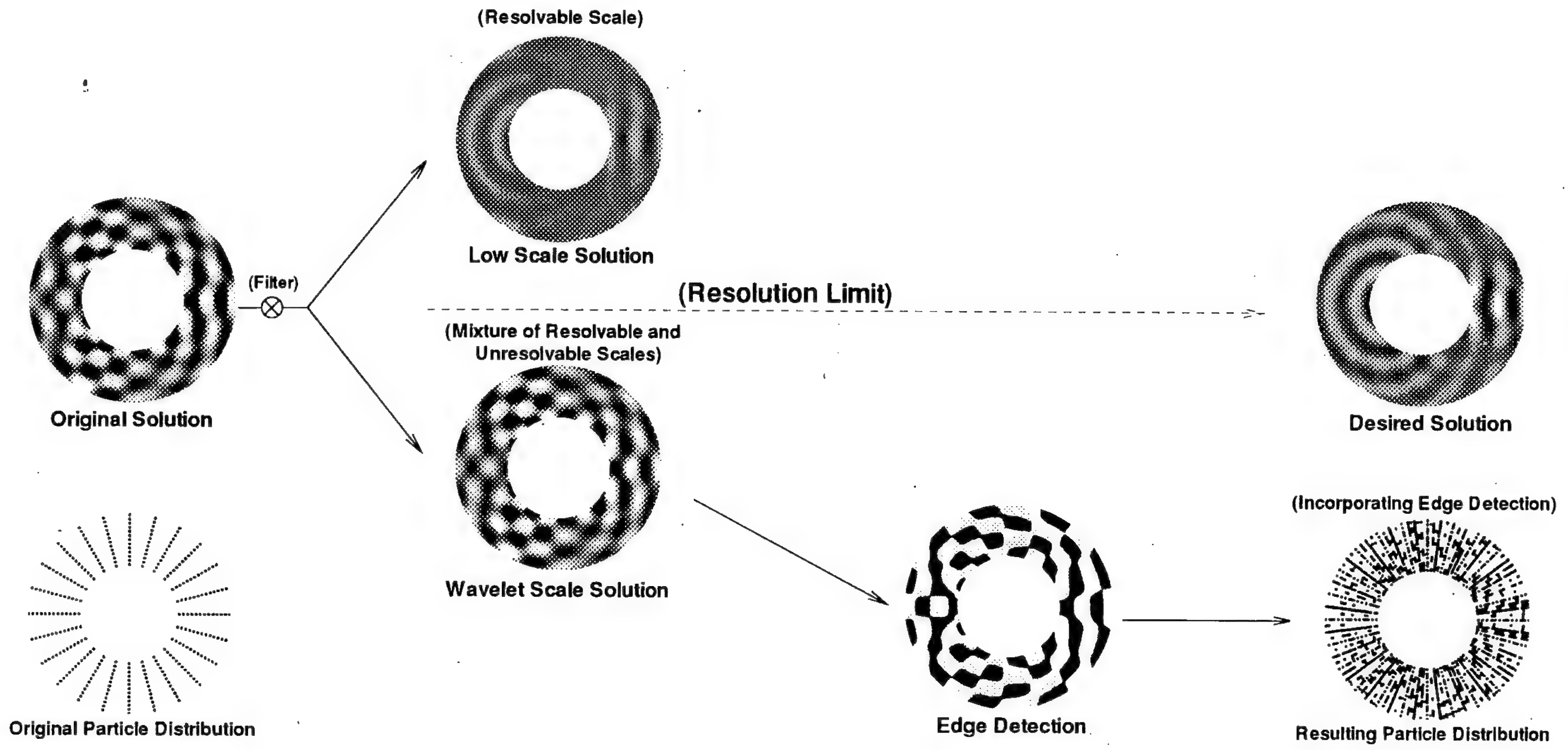


Figure 9.7 Multiple scale analysis for structural acoustics

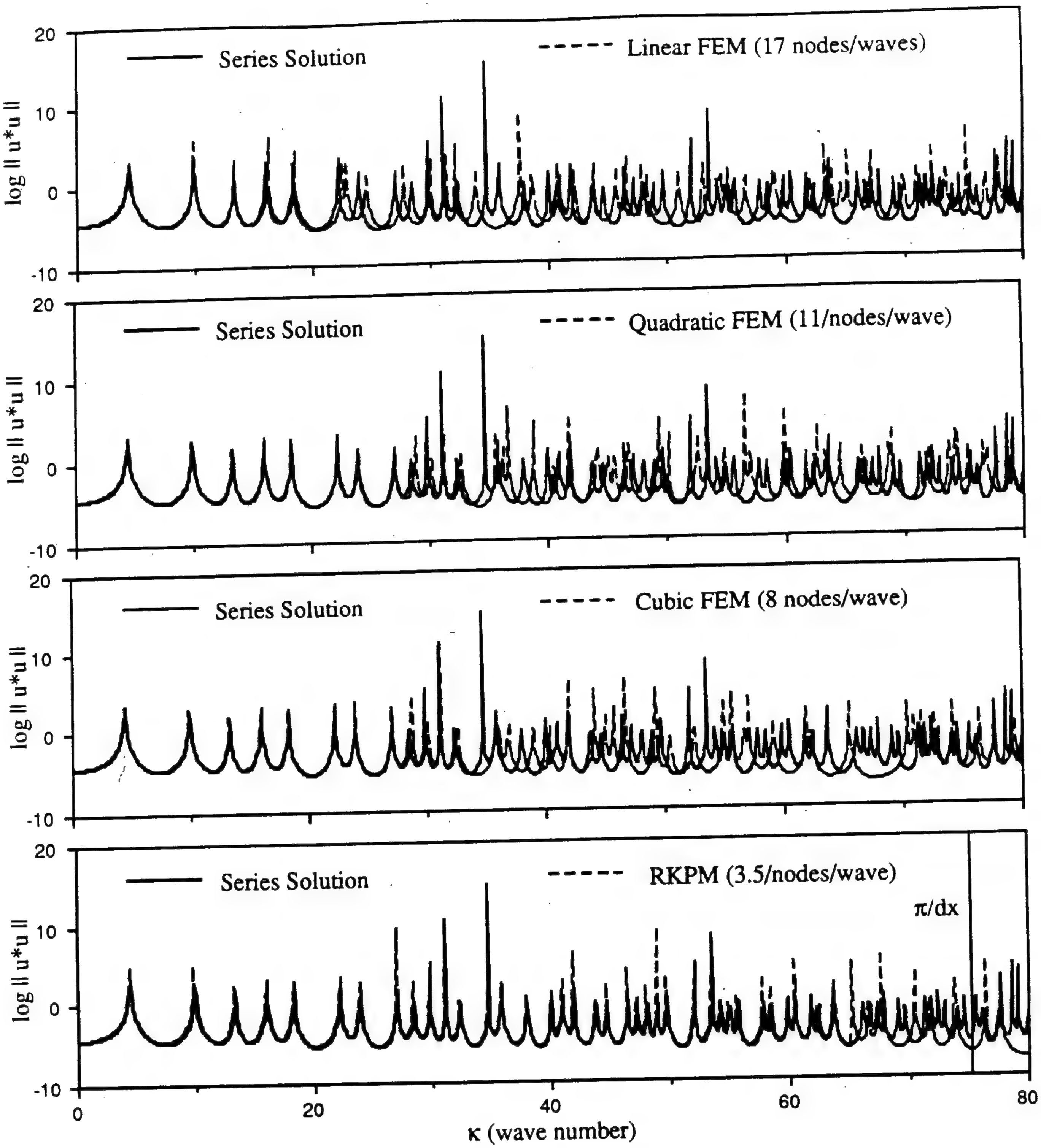


Figure 9.8b Comparison of computed resluts using finite element methods and RKPM wih series solution

CONCLUSIONS

The reproducing kernel particle method (RKPM) was reformulated in the framework of wavelet theories which include such concepts as reproducing conditions, discrete convolutions, and multiresolution analysis. It is shown that for a sufficiently smooth function, the interpolant expansion in terms of sampled values will converge to the original function in Sobolev norm. With this construction of interpolation error, the convergence rate can be measured in terms a new control variable, the dilation parameter of the window function.

By studying the convolution of spline family with polynomials, we are able to construct a new family of compactly supported scaling functions which can reconstruct polynomials of arbitrary order exactly. It was first shown that the interpolant error estimate is proportional to the wavelet solution. We have also implemented the first application of wavelet solution and edge detection in designing hp adaptive algorithm.

RKPM has been applied to such areas as structural acoustics, elastic-plastic deformation, computational fluid dynamics. RKPM is shown to provide an accurate mesh free algorithm and possess superior convergence. RKPM is envisioned as a new meshless method with completely different mentality, which will build a bridge between the traditional interpolation method and the spectral method. This is very desirable for the next generation of meshless methods.

ACKNOWLEDGMENT

The support of this research by Air Force Office of Scientific Research (AFOSR) is gratefully acknowledged. The under water bubble explosion problem as well as structural acoustics problem were partially supported by Office of Naval Research (ONR). The authors would like to thank Mr. Wei Hao, Mr. Sukky Jun, and Mr. Shaofan Li for their contributions in the preparation of this manuscript.

REFERENCES

Attaway, S. W., Heinstein, M. W., Mello, F. J. and Swegle, J. W., "Coupling of Smooth Particle Hydrodynamics With Pronto", preprint, (1993).

Babuska, I. and Melenk, J.M., "The Partition of Unity Finite Element Method." *Univ. of Maryland, Technical Note BN-1185*, (1995).

Beklkin, G. R., Coifman, I., Daubechies, S., Mallat, Y., Meyer, L., Raphael, and Ruskai, B., (eds) *Wavelets and Their Applications*, Cambridge, MA, (1992).

Belytschko, T., Lu, Y. Y. and Gu, L., "Element Free Galerkin Methods", *International Journal for Numerical Methods in Engineering*, **37**, 229-256, (1994a).

Belytschko, T., Lu, Y. Y. and Gu, L., "Fracture and crack growth by element free Galerkin methods", *Modelling Simulation Science Engineering*, **2**, 519-534, (1994b).

Belytschko, T., Lu, Y. Y., Gu, L. and Tabbara, M., "Element-Free Galerkin Methods for Static and Dynamic Fracture", *International Journal Solids Structures,* Vol. 32, No. 17/18, , pp. 2547-2570, (1995a).

Belytschko, T., Krongauz, Y., Fleming, M., Organ, D. and Liu, W. K., "Smoothing and Accelerated Computations in the Element Free Galerkin Method", submitted to *Journal of Computational Applied Mathematics*, (1995b).

Chui, C. K., *An Introduction to Wavelets*, Academic Press, (1992).

Chen, Y. J., Liu, W. K., C. T. Chang and R. A. Uras, "Multiresolution Reproducing Kernel Particle Method for Structural Acoustics", submitted to *Journal of Computational Acoustics*, (1995).

Daubechies, I., *Ten Lectures on Wavelets*, CBMS/NSF Series in Applied Mathematics, No. 61, SIAM Publication, (1992).

Duarte, C.A. and Oden, J.T., "*Hp* Clouds - A Meshless Method to Solve Boundary-Value Problems," *TICAM Report 95-05*, (1995).

Gingold, R. A. and Monaghan, J. J., "Smoothed Particle Hydrodynamics: Theory and Application to Non-Spherical Stars", *Mon. Not. Roy. Astron. Soc.*, **181**, 375-389, (1977).

Johnson, G.R., Peterson, E.H. and Stryrk, R.A., "Incorporation of an SPH Option into the EPIC code for a Wide Range of High Velocity Impact Computations," preprint, (1993).

Libersky, L. and Petschek, A.G., "Smooth Particle Hydrodynamics with Strength of Materials," *Proceedings of The next Free-Lagrange Conference,* Jackson Lake Lodge, Moran, Wyoming, June3-7, (1990).

Liu, W. K., Zhang, Y. and Ramirez, M. R., "Multiple Scale Finite Element Methods," *International Journal for Numerical Methods in Engineering*, **32**, 969-990, (1991).

Liu, W. K., and Haeussermann, U., "Multiple Temporal and Spatial Scale Methods," *New Methods in Transient Analysis,* P. Smolinski, W. K. Liu, G. Hulbert and K. Tamma, eds., PVP 246/AMD 143, ASME, pp.51-64, (1992).

Liu, W. K., and Oberste-Brandenburg, C., "Reproducing Kernel and Wavelet Particle Methods," *Aerospace Structures: Nonlinear Dynamics and System Response*, Eds. Cusumano, J. P., Pierre, C., and Wu, S. T., AD 33, ASME, pp. 39-56, (1993a).

Liu, W. K., Adee, J., and Jun, S., "Reproducing Kernel Particle Methods for Elastic and Plastic Problems," *Advanced Computational Methods for Material Modeling*, Eds. Benson, D. J. , and Asaro, R. A., AMD 180 and PVP 268, ASME, pp. 175-190, (1993b).

Liu, W. K., and Chen, Y. J., "Wavelet and Multiple Scale Reproducing Kernel Methods," *International Journal for Numerical Methods in Fluids*, vol. 21, pp. 901-932, (1995).

Liu, W. K., "Feature Article on An Introduction to Wavelet Reproducing Kernel Particle Methods," USACM Bulletin, Vol. 8, No. 1, pp.3-16, (1995).

Liu, W. K., Jun, S., Li, S., Adee, J., and Belytschko, T., "Reproducing Kernel Particle Methods for Structural Dynamics," *International Journal for Numerical Methods in Engineering*, vol. 38, pp. 1655-1679, (1995a).

Liu, W. K., Jun, S. and Zhang, Y. F., "Reproducing Kernel Particle Methods", *International Journal of Numerical Methods in Fluids*, vol. 20, pp. 1081-1106, (1995b).

Liu, W. K., Chen, Y., Jun, S., Chen, J. S., Belytschko, T., Pan, C., Uras, R. A. and Chang, C. T., "Overview and Applications of the Reproducing Kernel Particle Methods," *Archives of Computational Methods in Engineering, State of the Art Reviews*, accepted for publications, September, (1995c).

Liu, W. K., Jun, S., Sihling, D. T., Chen, Y. and Hao, W., "Multiresolution Reproducing Kernel Particle Method for Computational Fluid Dynamics", presented at the Third US-Japan Symposium on Finite Element Methods in Large-Scale Computational Fluid Dynamics, (1995d).

Liu, W. K., Li, S. and Belytschko, T., "Moving Least Square Kernel Galerkin Method (I) Methodology and Convergence," submitted to *Computer Methods in Applied Mechanics and Engineering*, (1995e).

Liu, W. K., Chen, Y. and Uras, R. A., "Enrichment of the Finite Element Method With the Reproducing Kernel Particle Method," Current Topics in Computational Mechanics, Eds. J. F. Cory, Jr., and J. L. Gordon, ASME PVP-Vol. 305, pp. 253-258, (1995f).

Liu, W. K., Chen, Y. and Hao, W., "Adaptivity by Multiresolution Reproducing Kernel Methods" submitted to Computational Mechanics, (1996a)

Lu, Y. Y., Belytschko, T. and Gu, L., "A New Implementation of the Element Free Galerkin Method", *Computer Methods in Applied Mechanics and Engineering*, **113**, 397-414, (1994).

Mallat, S., "Multi-resolution Approximations and Wavelet Orthogonal Bases of L2(R)," *Trans. Amer. Math. Soc.,* 315, 69-87, (1989).

Monaghan, J. J., "An Introduction to SPH", *Comp. Phys. Comm.*, **48**, 89-96, (1988).

Nayroles, B., Touzot, G. and Villon, P., "Generalizing the Finite Element Method: Diffuse Approximation and Diffuse Elements," *Computational Mechanics*, Vol.10, pp.307-318, (1992).

Oñate, E., Idelsohn, S. and Zienkiewicz, O. C., "Finite Point Methods in Computational Mechanics", *International Center for Numerical Methods in Engineering*, (July, 1995).

Shodja, H. M., Mura, T., and Liu, W. K., "Multiresolution Analysis of a Micromechanical Model," *Computational Methods in Micromechanics*, ASME AMD 212/MD 62, eds. S. Ghosh and M. Ostoja-Starzewski, pp. 33-54, (1995).

Strang, G., "Wavelets and Dilation Equations: a Brief Introduction," *SIAM Review*, 31, 614-627, (1989).

Sulsky, D., Chen, Z. and Schreyer, H. L., "The Application of a Material-Spatial Numerical Method to Penetration," *New Methods in Transient Analysis*, eds. Smolinski, P., Liu, W. K., Hulbert, G. and Tamma, K., ASME, PVP Vol. 246/AMD Vol.143, pp.91-102, (1992).

Williams, J. R., and Amaratunga, K., "Introduction to Wavelets in Engineering," *International Journal for Numerical Methods in Engineering*, Vol. 37, 2365-2388, (1994).

Yagawa, G., Yamada, T. and Kawai, "Some Remarks on Free Mesh Method: A kind of Meshless Finite Element Methods," To be presented at ICES-95, (1995).

APPENDIX A 3D SHAPE FUNCTION AND DERIVATIVE

A1. Shape Function

For a linear basis in 3D, the reproducing equation is (c.f. Eq. 5.3.3):

$$u^{R_a}(\boldsymbol{x}) = \sum_{j=1}^{NP} u(\boldsymbol{x}_j) C_a(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_j)\boldsymbol{\Phi}_a(\boldsymbol{x}-\boldsymbol{x}_j)\Delta V_j \tag{A1.1}$$

where $\boldsymbol{\Phi}_a(\boldsymbol{x}-\boldsymbol{x}_j) \equiv \phi_{a_x}(x-x_j)\phi_{a_y}(y-y_j)\phi_{a_z}(z-z_j)$, ϕ_{a_x}, ϕ_{a_y}, ϕ_{a_z} are cubic spline functions, and ΔV_j is the nodal volume. The correction function $C_a(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_j)$ is defined as:

$$C_a(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_j) = b_0(\boldsymbol{x}) + b_1(\boldsymbol{x})(x-x_j) + b_2(\boldsymbol{x})(y-y_j) + b_3(\boldsymbol{x})(z-z_j) \tag{A1.2}$$

The correction function coefficients $\boldsymbol{b}(\boldsymbol{x})$ are computed by Eq. (5.3.6):

$$\boldsymbol{b}(\boldsymbol{x}) = \tilde{\boldsymbol{M}}^{-1}(\boldsymbol{a},\boldsymbol{x})\boldsymbol{P}(\boldsymbol{0}) \tag{A1.3}$$

where

$$\boldsymbol{P}(\boldsymbol{0}) = [1, 0, 0, 0]^T \tag{A1.4a}$$

$$\tilde{\boldsymbol{M}}(\boldsymbol{a},\boldsymbol{x}) = \begin{bmatrix} \tilde{m}_{000}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{100}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{010}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{001}(\boldsymbol{a},\boldsymbol{x}) \\ \tilde{m}_{100}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{200}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{110}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{101}(\boldsymbol{a},\boldsymbol{x}) \\ \tilde{m}_{010}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{110}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{020}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{011}(\boldsymbol{a},\boldsymbol{x}) \\ \tilde{m}_{001}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{101}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{011}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{002}(\boldsymbol{a},\boldsymbol{x}) \end{bmatrix} \tag{A1.4b}$$

and

$$\tilde{m}_{IJK}(\boldsymbol{a},\boldsymbol{x}) = \sum_{j=1}^{NP} (x-x_j)^I (y-y_j)^J (z-z_j)^K \boldsymbol{\Phi}_a(\boldsymbol{x}-\boldsymbol{x}_j)\Delta V_j \tag{A1.4c}$$

The shape function is then given as:

$$\begin{aligned} N_j(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_j) &= C_a(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_j)\boldsymbol{\Phi}_a(\boldsymbol{x}-\boldsymbol{x}_j)\Delta V_j \\ &= \left[b_0(\boldsymbol{x}) + b_1(\boldsymbol{x})(x-x_j) + b_2(\boldsymbol{x})(y-y_j) + b_3(\boldsymbol{x})(z-z_j)\right]\boldsymbol{\Phi}_a(\boldsymbol{x}-\boldsymbol{x}_j)\Delta V_j \end{aligned} \tag{A1.5}$$

A2. First Derivative of Shape Function

The first derivative of shape function is given as:

$$N_{j,x}(\boldsymbol{x}:\boldsymbol{x}-\boldsymbol{x}_j)=\left[C_{a,x}(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_j)\boldsymbol{\Phi}_{\boldsymbol{a}}(\boldsymbol{x}-\boldsymbol{x}_j)+C_a(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_j)\boldsymbol{\Phi}_{\boldsymbol{a},x}(\boldsymbol{x}-\boldsymbol{x}_j)\right]\Delta V_j \tag{A2.1}$$

or in matrix form:

$$\begin{bmatrix} N_{j,x} \\ N_{j,y} \\ N_{j,z} \end{bmatrix}=\begin{bmatrix} C_{a,x}(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_j)\boldsymbol{\Phi}_{\boldsymbol{a}}(\boldsymbol{x}-\boldsymbol{x}_j)+C_a(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_j)\boldsymbol{\Phi}_{\boldsymbol{a},x}(\boldsymbol{x}-\boldsymbol{x}_j) \\ C_{a,y}(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_j)\boldsymbol{\Phi}_{\boldsymbol{a}}(\boldsymbol{x}-\boldsymbol{x}_j)+C_a(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_j)\boldsymbol{\Phi}_{\boldsymbol{a},y}(\boldsymbol{x}-\boldsymbol{x}_j) \\ C_{a,z}(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_j)\boldsymbol{\Phi}_{\boldsymbol{a}}(\boldsymbol{x}-\boldsymbol{x}_j)+C_a(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_j)\boldsymbol{\Phi}_{\boldsymbol{a},z}(\boldsymbol{x}-\boldsymbol{x}_j) \end{bmatrix}\Delta V_j \tag{A2.2}$$

where

$$C_{a,x}(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_j)=b_{0,x}(\boldsymbol{x})+b_{1,x}(\boldsymbol{x})(x-x_j)+b_{2,x}(\boldsymbol{x})(y-y_j)+b_{3,x}(\boldsymbol{x})(z-z_j)+b_1(\boldsymbol{x}) \tag{A2.3a}$$

$$C_{a,y}(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_j)=b_{0,y}(\boldsymbol{x})+b_{1,y}(\boldsymbol{x})(x-x_j)+b_{2,y}(\boldsymbol{x})(y-y_j)+b_{3,y}(\boldsymbol{x})(z-z_j)+b_2(\boldsymbol{x}) \tag{A2.3b}$$

$$C_{a,z}(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_j)=b_{0,z}(\boldsymbol{x})+b_{1,z}(\boldsymbol{x})(x-x_j)+b_{2,z}(\boldsymbol{x})(y-y_j)+b_{3,z}(\boldsymbol{x})(z-z_j)+b_3(\boldsymbol{x}) \tag{A2.3c}$$

The first derivative of the correction function coefficients are obtained as follows:

$$\boldsymbol{b}_{,x}(\boldsymbol{x})=-\tilde{\boldsymbol{M}}^{-1}(\boldsymbol{a},\boldsymbol{x})\tilde{\boldsymbol{M}}_{,x}(\boldsymbol{a},\boldsymbol{x})\boldsymbol{b}(\boldsymbol{x}) \tag{A2.4}$$

where

$$\tilde{\boldsymbol{M}}_{,x}(\boldsymbol{a},\boldsymbol{x})=\begin{bmatrix} \tilde{m}_{000,x}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{100,x}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{010,x}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{001,x}(\boldsymbol{a},\boldsymbol{x}) \\ \tilde{m}_{100,x}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{200,x}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{110,x}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{101,x}(\boldsymbol{a},\boldsymbol{x}) \\ \tilde{m}_{010,x}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{110,x}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{020,x}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{011,x}(\boldsymbol{a},\boldsymbol{x}) \\ \tilde{m}_{001,x}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{101,x}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{011,x}(\boldsymbol{a},\boldsymbol{x}) & \tilde{m}_{002,x}(\boldsymbol{a},\boldsymbol{x}) \end{bmatrix} \tag{A2.5a}$$

and

$$\tilde{m}_{IJK,x}(\boldsymbol{a},\boldsymbol{x})=\sum_{j=1}^{NP}\left[(x-x_j)^I(y-y_j)^J(z-z_j)^K\,\boldsymbol{\Phi}_{\boldsymbol{a}}(\boldsymbol{x}-\boldsymbol{x}_j)\right]_{,x}\Delta V_j \tag{A2.5b}$$